Title Strategies for Neural Networks in Ballistocardiography with a View Towards Hardware Implementation

Name Xinsheng Yu

STRATEGIES FOR NEURAL NETWORKS IN BALLISTOCARDIOGRAPHY WITH A VIEW TOWARDS HARDWARE IMPLEMENTATION

By

Xinsheng Yu

A thesis submitted for the degree of
Doctor of Philosophy
at the University of Luton

Faculty of Design and Technology

November, 1996

ABSTRACT

The work described in this thesis is based on the results of a clinical trial conducted by the research team at the Medical Informatics Unit of the University of Cambridge, which show that the Ballistocardiogram (BCG) has prognostic value in detecting impaired left ventricular function before it becomes clinically overt as myocardial infarction leading to sudden death. The objective of this study is to develop and demonstrate a framework for realising an on-line BCG signal classification model in a portable device that would have the potential to find pathological signs as early as possible for home health care.

Two new on-line automatic BCG classification models for time domain BCG classification are proposed. Both systems are based on a two stage process: input feature extraction followed by a neural classifier. One system uses a principal component analysis neural network, and the other a discrete wavelet transform, to reduce the input dimensionality. Results of the classification, dimensionality reduction, and comparison are presented. It is indicated that the combined wavelet transform and MLP system has a more reliable performance than the combined neural networks system, in situations where the data available to determine the network parameters is limited. Moreover, the wavelet transform requires no prior knowledge of the statistical distribution of data samples and the computation complexity and training time are reduced. Overall, a methodology for realising an automatic BCG classification system for a portable instrument is presented.

A fully parallel neural network design for a low cost platform using field programmable gate arrays (Xilinx's XC4000 series) is explored. This addresses the potential speed requirements in the biomedical signal processing field. It also demonstrates a flexible hardware design approach so that an instrument's parameters can be updated as data expands with time. To reduce the hardware design complexity and to increase the system performance, a hybrid learning algorithm using random optimisation and the backpropagation rule is developed to achieve an efficient weight update mechanism in low weight precision learning. The simulation results show that the hybrid learning algorithm is effective in solving the network paralysis problem and the convergence is much faster than by the standard backpropagation rule. The hidden and output layer nodes have been mapped on Xilinx FPGAs with automatic placement and routing tools. The static time analysis results suggests that the proposed network implementation could generate 2.7 billion connections per second performance.

TO THE MEMORY OF MY PARENTS

ACKNOWLEDGEMENTS

Firstly, I would like to thank all those who have given me the strength and encouragement to carry on this work in such separated periods. I would like to thank my supervisors, Don Dent, Colin Osborn, and Dr. David Crecraft in his function as external supervisor, for their support and discussion over the three years. I am particularly grateful to the people who provided me their published and unpublished works, answered my questions, and gave creative comments, who made this research certainly original.

I am greatly indebted to Dr. Rudolf Hanka, who encouraged me to explore the Ballistocardiography (BCG) analysis system on my first visit to the Medical Informatics Unit in the University of Cambridge. The established collaboration has laid the base for most of the work presented in this thesis. Thanks also go to Dr. Gonjong Zhao, who unselfishly explained to me his unpublished works and patiently answer my naive questions about the BCG. I owe thanks to Dr. Roger Harvey for his consistent support and expertise. I thank Colin Osborn for proof reading this thesis to eradicate my Chinesisms. Thanks also to David Crecraft and Don Dent for proof reading several chapters.

I thank Steve Mortimer for taking me to London for "appeal hearing", which added an extra experience to a PhD study. Thanks are also due to Dr. Robert Hearing for smoothing the daily activities, and to David Moon for his helping at the Lab. I thank all staff members at the Faculty of Design and Technology and 'The Research Centre' for their support. I also direct my special thanks to Val Newman for her day by day assistance at 'The Research Centre'.

I tug my forelock to the members of the Institute of Oceanology; Prof. Dejun Gong, Prof. Qingyu Shan, and Yan Li, who never failed to give me support. I also express my warm appreciation to Dr. Malcom Green for his understanding and supporting of my decision to pursue the degree study.

There is so much more to a PhD than simply the numbers and programming. The friendship, excited entertainment and stimulating discussions provided by my friends and 'Yellow team-mate' will never pass from my memory. It has made those up and down days pass by so quickly. I only realise this now that I am writing up and all those times will be a nice past memory. Thanks also to fellow students for creating an inspiring atmosphere in Luton. Thanks to Jonathan Caplan and Chaomei Peng with whom I have shared 'extremely high voltage shocks' at the "Electronics Lab". Many thanks are also due to Luton Chinese Colleagues who have made tremendous '*GAME BALLS*'.

I thank all members of my family, for providing enormous support during my time in England. My biggest thanks is due to my dear wife, Li Hua, and our lovely Bob for their tolerance, faith and love. I owe recompense for all those days, that much of the present work was accomplished at the expense of missing them. I am assured that they will share my satisfaction at the completion of this work.

I acknowledge the help of the University of Luton and Faculty of Design and Technology who provide funding for maintenance and for attending conferences. I also acknowledge the Institute of Oceanology, Chinese Academy of Sciences and the Institute of Electrical Engineers, UK for financial support of presenting papers in conferences.

DECLARATION

The research described in this thesis was carried out by the author during February 1993 to April 1994, and January 1995 to November 1996 respectively. Except the material drawn from other sources which is acknowledged in the text, the work presented in the thesis is believed to be original and is not the result of work done in collaboration. No part of this thesis has been submitted to any other University for any degree, diploma or other qualification.

Xinsheng Yu

November, 1996

Contents

List of Figures

List of Tables

Abbreviations

ANNs	Artificial Neural Networks
ANSI	American National Standards Institute
ART	Adaptive Resonance Theory
ASICs	Application Specific Integrated Circuits
BCG	Ballistocardiography
BCPS	Billion Connections per second
BP	Backpropagation
CAD	Computer Aided Design
CLA	Carry Lookahead Adder
CLBs	Configurable Logic Blocks
CMOS	Complementary Metal-Oxide-Semiconductor Transistor
CPLDs	Complex Programmable Devices
CPS	Connection Per Second
CSA	Carry Save Adder
CT	Computerised Tomagraphy
CWT	Continuous Wavelet Transformation
DC	Direct Coupled
DNNA	Digital Neural Network Architecture
DRAM	Dynamic Random Access Memory
DSP	Digital Signal Processing
DTW	Dynamic Time Wrapping
DWT	Discrete Wavelet Transformation
ECG	Electrocardiograph
EEG	Electroencephalogram
EPROM	Erasable Programmable Read Only Memory
ET	Error Threshold
FIR	Finite Impulse Response
FPGAs	Field Programmable Gate Arrays
GT	Gradient Threshold
HDL	Hardware Description Language
HF	High Frequency
I/O	Input/Output

ICD	Implantable Cardioverter Defibrillator
IOBs	Input/Output Blocks
K-L	Karhunen-Loeve
K-Y	Kittler-Young
LCA	Logic Cell Array
LF	Low Frequency
LVQ	Learning Vector Quantization
MCPS	Million Connections Per Second
MHz	Mega Hertz
MLP	Multilayer Perceptron
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
MUSIC	Multiprocessor System with Intelligent Communication
PC	Personal Computer
PCA	Principal Component Analysis
PDP	Parallel Distributed Processing
PPR	Partition, Place and Route
PPS	Patterns Per Second
PQFP	Plastic Quad Flat Package
PROM	Programmable Read Only Memory
QGDR	Quantized Gradient Descent Rule
QMF	Quadrature Mirror Filter
RAM	Random Access Memory
RBF	Radial Basis Functions
RCA	Ripple Carry Adder
REM	Rapid Eye Movement
ROM	Read Only Memory
RPMs	Relational Placed Macros
SCG	Seismocardiography
SCSB	Static Charge Sensitive Bed
SOM	Self Organising Map
SRAM	Static Random Access Memory
ULF	Ultra Low Frequency
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
VLSI	Very Large Scale Integration
XNF	Xilinx Netlist Format
XOR	Exclusive OR

Chapter 1

Introduction

1.1 Introduction

Heart disease is a major public health problem in the industrialised countries. According to research in Glasgow (Dargie, 1994), the proportion of heart failure caused deaths was 55% in 5 years in the hospital, and 40% to 60% in 1 year outside the hospital. Figure 1.1 also shows the results of survey data in the United States in 1991 (Cura, 1995) which indicated that heart disease is the main killer both in men and women. Reduction in the mortality of this disease could be achieved primarily through early detection and treatment. Over the past few decades, scientists and doctors have built up many methods to monitor cardiac performance and to diagnose heart disease. Among these methods, the non-invasive approaches for monitoring physiological values, such as Electrocardiography (ECG), Seismocardiography (SCG), and Ballistocardiography (BCG), have continued to stimulate interest.

Traditionally, the physicians in hospitals need to interpret characteristics of the measured records and to calculate relevant parameters to determine whether or not the heart shows signs of cardiac disease. Recently, the advances in computer and electronic technology have provided a basis for automatic cardiac performance monitoring and heart disease diagnosis to assist clinical practice by saving diagnostic time. These technologies along with artificial intelligence research have also established entirely new applications in detecting the risk of heart disease (Harrison et al., 1991). However, the demands of practical health care require that these technologies are not limited to hospital environments but are able to detect the vital signs of heart disease during daily life under unrestricted conditions. For example, it would be helpful for both doctors and patients, if the preliminary heart condition could be monitored regularly at the home before making the decision whether or not it is necessary to visit hospital, for further multiple assessment and treatment. Thus, the time and transport expenses could be saved. Moreover, the recent state-of-the-art communications technology has influenced the new technology, telemedicine, which lets doctors investigate the patients via electronic information exchange. Complex equipment is of little use to this application unless it simplifies the complicated procedure and is less expensive (Ream, 1978). These requirements lead to increasing innovation in portable systems resulting in reduction in size, weight and power consumption to an extent that the instruments are available in routine life monitoring and diagnosis to many users. Wiesspeiner and Xu (1992) reported a miniature data processing system for monitoring circulation related biosignals. Bertolet and Pepine (1994) described a portable device for ECG monitoring for which the recorded signal can be transmitted to the hospital via a telephone line. Leong and Jabri (1995) implemented an artificial neural network algorithm to an implantable cardioverter defibrillator (ICD) device for arrhythmia classification. The advantage of these devices is that patients are allowed to monitor their heart condition in a variety of less restricted situations. It offers a great flexibility in real world applications.

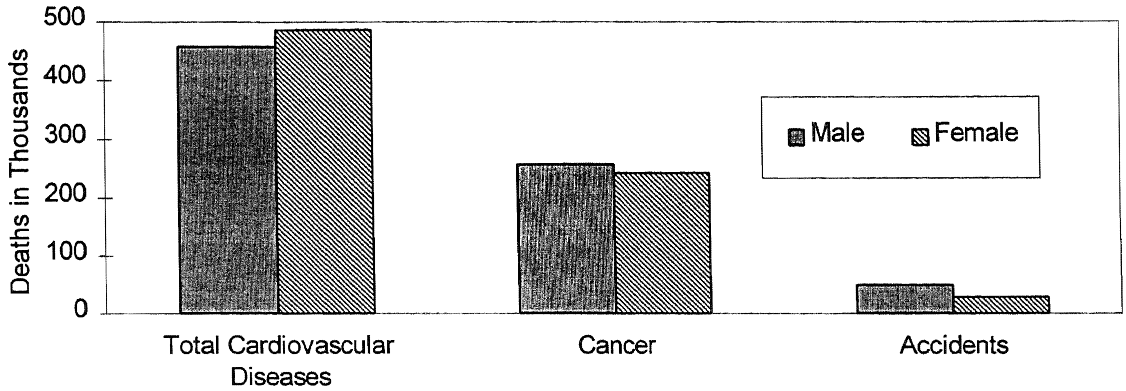


Figure 1.1: Heart disease is the greatest killer as shown by data for 1991 in U. S. A., after Cura (1995)

A ballistocardiograph is a record of the movements of the body caused by shifts in the centre of mass of the blood, including recoil force during cardiac ejection, and by the motion of the heart itself during each cardiac cycle (Star and Noordergraaf, 1967). This body movement can be recorded by a BCG apparatus provided that the coupling of the subject to the measurement platform is sufficiently tight. There are three types of BCG apparatus. In the first the subject lies on a bed. In the second the subject sits on a chair, and in the third the subject stands on a platform. The main distinguishing feature of this technique is that no electrodes are needed to be attached to the body during the measurements. This provides a potential application that the BCG can serve as a relatively low cost, non-invasive, easy-to-use, home screening procedure for heart condition assessment so saving transport, time and money.

In the past several years, the BCG has been used in a variety of clinical studies such as prognosis, monitoring, screening, physical conditioning, stress tests, evaluation of therapy, cardiovascular surgery and clinical conditions (Brown et al., 1965; Starr and Noordergraaf, 1967; Marineli et. al., 1991). Starr and Wood (1961) studied the ballistocardiograms of a group of volunteers and followed them for 17-23 years until some of them eventually developed various kinds of cardiac disease. Their results have shown that , in spite of the difficulties in calibration, a simple qualitative classification of the resting BCG could be used to predict, with considerable confidence, the development of ischemic myocardial disease, even when the electrocardiogram was normal. Other research also proved the prognostic value of the BCG in cardiovascular screening and epidemiological studies (Kiessling, 1970; Lynn and Wolf, 1974). Using BCG for sleep research has also been conducted (Erkinjuntti et al., 1984; Jansen and Shankar, 1993; Polo et al. 1992). In a recent pilot study, Zhao (1994) has used combined statistical pattern recognition and neural network methods for BCG signal analysis and the preliminary results indicated that the chair BCG (subject sitting on a chair) signal has potential application in the prognosis of heart attack risk. It would therefore be of value if the BCG could be used as a non-invasive alternative for the prediction of subjects at risk of heart attack that could help clinicians to establish an early treatment.

Although studies have been made in the last few decades on the subject of improving BCG measurement technology, there has tended to be relatively little work done to attempt to improve the BCG process and analysis methods by using modern signal processing technology (Zhao, 1994). In fact, the analysis is usually performed by manually taking the measurements from a chart recorder or on digital computers with software packages. There is little work on developing computer assisted analysis systems that are capable of operating in real time, especially those incorporating artificial intelligence methods for BCG pattern classification. Moreover, no reports have been made in the literature to implement decision making algorithms in a portable system for real world application (Yu et al., 1995). There is a need to add such capabilities in the field of the BCG, in particular, the problems of diagnosis and prognosis, so that the BCG data can be processed in an integrated environment. Furthermore, the automated BCG diagnostic system can provide a means for BCG diagnostic standardisation. This thesis is concerned with developing and demonstrating the effectiveness of an automatic BCG classification system which can be implemented using artificial neural networks architecture to provide an inexpensive and convenient diagnosis and prognosis means in patients' homes. The BCG data used in this study is provided by the Medical Informatics Unit of the University of Cambridge on a collaboration basis.

1.2 Ballistocardiography and Ballistocardiography Instrumentation: An Overview

1.2.1 BCG Background

The ballistocardiographic system as a whole consists of the cardiovascular system as primary source of force, the human body acting as a connecting link, and the mechanical part of the ballistocardiographic apparatus as the receiver of the mechanical transmission system. The transmission of the force from the source to the ballistocardiographic apparatus is influenced by the mechanical characteristics of all these parts (Smith, 1974). The function of a BCG transducer is therefore to transform the BCG force into a form that can be readily available for direct analysis or acquisition.

In contrast to the ECG, the BCG signal is a number of events which act simultaneously throughout the cardiac cycle, so that a wave tip merely means that the sum of several events is maximal or minimal. Therefore, a wave peak or inflexion of the record cannot, in general, be attributed to a single physiologic event in clinical diagnosis, although one event may be the major contributor (Smith, 1974). Figure 1.2 presents a typical cycle of a BCG signal and simultaneous ECG records. It is a difficult signal to decompose. Usually the R-wave of the ECG record is used to identify the BCG cardiac cycles. The pattern begins with a small headward deflection , the H wave, which is thought to be initiated by the apex thrust that occurs during the isometric contraction (Hamilton et al., 1945). The I wave is probably caused by the footward component of the cardiac recoil accompanying ventricular ejection. The J wave corresponds to the deceleration of either the blood or impulse wave, or both, by the aortic and pulmonary arches and possibly by the head (Brown et al., 1952). The J wave is the highest normal deflection and estimates the stroke volume and the forcefulness of cardiac ejection (Brown et al. 1952). The J-K deflection probably represents the deceleration of the footward travelling impulse wave by the arteriolar peripheral resistance in the legs (Hamilton et al., 1945). The L, M, N, and O waves are called "after waves" which are produced during diastole (Polo, 1992).

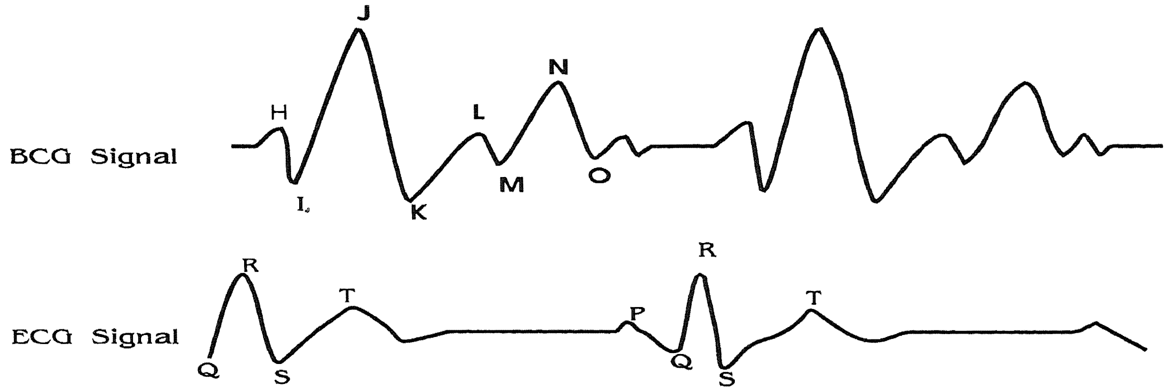


Figure 1.2: A cardiac cycle of an BCG and ECG signal

The BCG was first observed by Parry in 1786 to detect the oscillations of the trunk synchronous with heart beat. The first BCG recording was published by Gordon (1877) who demonstrated that motion of the body occurs with each heart beat. Along with Cougus and Haldane, Henderson constructed a much simpler and less reliable instrument to record BCG. Starr and his co-workers (1939) paved the way for cardiac evaluation based on analysis of the BCG by showing that the ejection from the right ventricle reflected in the BCG is approximately one-fifth that of the left ventricle. Thus, they proved that the BCG is mainly determined by the ejection pattern of the left ventricle of the heart. Starr (1939) also established a model of cardiac dynamics. He judged that the deflection of a high frequency BCG waveform was caused by the mass of the blood and its acceleration. Since then scattered attempts have been made to investigate the BCG and its application in medicine. In 1970, the emphasis on non-invasive techniques for the cardiac diagnosis in contemporary medicine has made the ballistocardiogram more appealing (Smith, 1974). The development in recent years has involved new methods for measuring and interpreting ballistocardiograms based on transducer and computer techniques (Alihanka et al., 1981; Hanka, 1979; Rohen, 1985; Schwerdt et al., 1987; Zhao, 1994). These developments allow measurements to be standardised and to be more accurate.

The type of apparatus used in obtaining ballistocardiograms vary in their frequency response and structure. They can be divided into high frequency (HF) BCG with natural system frequency around 15 Hz , Low frequency (LF) BCG with natural system frequency about 1.0 to 1.5 Hz and Ultra Low frequency (ULF) BCG with a natural frequency below 0.5 Hz. However, there is no specific definition of frequency ranges for these systems in the literature.

It was noted that both the HF BCG and the LF BCG are more easily affected by high frequency interference than ULF BCG (Noordergaaf, 1965). The ULF BCG has proved to be more useful and studied more in modern BCG applications (Wright, 1975; Alihanka, 1987; Zhao, 1994). It is the more practical system used in today's BCG study.

1.2.2 Survey of Existing BCG Instrumentation

Wright (1975) described a practical portable instrument to record the BCG. The primary goal of his work was to develop a practical BCG recording instrument for following changes in heart action in the intensive care situation or during anaesthesia, when the patient will normally be lying fairly still. The instrument consists of a pneumatic accelerometer which is worn on the top of the head, secured by a strap under the chin. The recorded BCG waveform can be displayed on a meter, or interfaced to a BCG recorder, to record the waveform for later analysis. This method provides the potential application for the BCG signal to be recorded in the patient's bed at home. However, there is no signal processing part included in the instrumentation. The portable BCG device provides an interface port which can be connected to either a chart recorder or a data acquisition system to record the signal for later analysis. However, the BCG analysis must be done in the hospital or laboratories. Therefore, it is an on-line recording instrument and needs off-line analysis equipment.

Alihanka (1987) developed a static charge sensitive bed (SCSB) to measure the BCG signals. The SCSB provides a new method that allows long-term monitoring of the BCG in the subject's own bed. The BIOREC Company has now made a prototype as a commercial product based on the SCSB method for recording the BCG signal and body movement during sleep, which is named BioMatt Monitoring System (BIOREC, 1994). In this method, the subject lies on a mattress system and various body movements induce potential difference between two large metal plates located under the mattress and isolated from each other by a stiff insulating plate. The potential difference between the metal plates is recorded by a conventional differential amplifier. The recorded signal is a mixed effect of body movements, heart beat, blood transportation and respiration etc., that all contribute to the change of the static charge distribution. The SCSB method provides a simple and comfortable way of BCG detection in the home. A software package runs on a personal computer (PC) to record, process and monitor on-line SCSB output. The software can automatically calculate three parameters: respiratory movement, cardiac force and body movement. However, the on-line monitoring system, as shown in Figure 1.3, needs a 386 or later version PC with the minimum hardware requirements of at least 100 MB hard disk space for saving the original signals and 10 MB free hard disk space to run the specific program. It is an automated monitoring and data acquisition system. The diagnosis decisions still rely on doctors' knowledge.

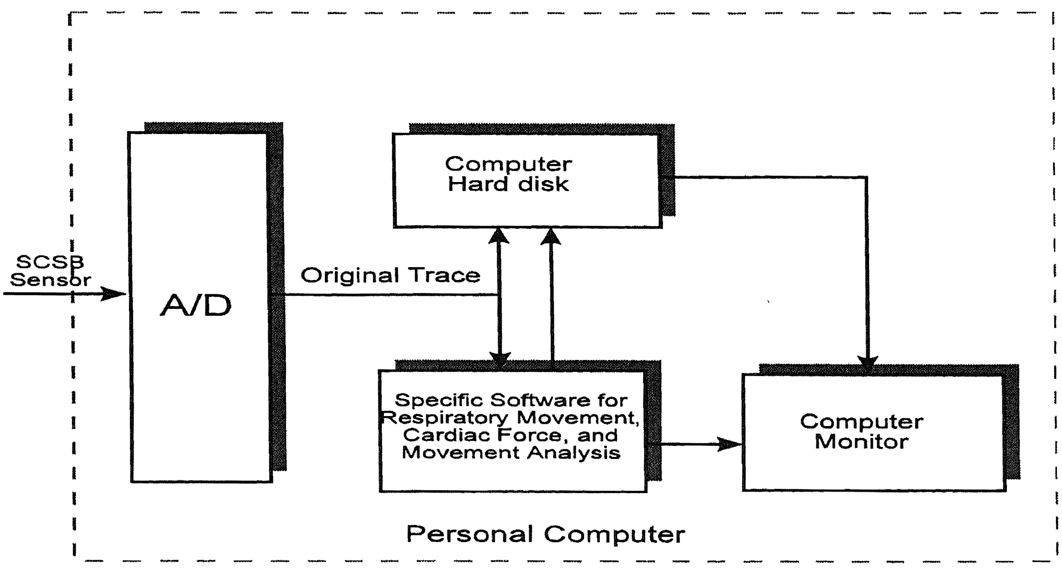


Figure 1.3: An on-line monitoring system for Static Charge Sensitive Bed (SCSB).

In order to obtain a three dimensional impression of ballistocardiac action, Schwerdt et al. (1987) developed a computer based force vectorballistocardiography that detects the ballistocardiogram in three directions. Three piezoelectric force transducers are placed on the patient who lies on the bed. Each of the transducers permits simultaneous recording of dynamic forces in three orthogonal directions. The outputs of the x, y and z components of each force transducer are added algebraically to give the total x, y and z components of the BCG force vectors respectively. A computer is used to co-ordinate three channel sampling. The authors only described the simple signal processing procedure implemented on the computer for noise removal. A reliable method for automatic analysis of the vectorballistocardiography signals is required for real application. The instrumentation can be used as an on-line monitoring application.

In order to develop a more convenient, effective and comfortable method for the detection of ballistocardiograms, Rohen (1985) has developed a light-weight chair with a soft mattress that replaces the rigid surface of the chair to measure the BCG signal. Recently, a new design was carried out by Zhao (1994). The new design is based on the mechanism of displacement measurement with a capacitive transducer. The basic idea is that conductor plates or electrodes embedded in the mattress can form a capacitor, the capacitance of which varies as the distance between plates is changed by the force (human body weight, BCG force, etc.) acting upon the mattress. The BCG signal was measured from the seat of the chair and the surface ECG is recorded from the electrodes at the arms of the chair. The BCG signal processing procedure is shown in Figure 1.4. The BCG signals were logged on to a microcomputer on site and the data files were later transferred to the laboratory for analysis on a PC. The BCG signal pre-processing method discussed in section 2.4.2 of chapter 2 was used. The BCG signal analysis was based on a combination of a Kittler-Young transform (Kittler and Young, 1972) and a multilayer perceptrons.

The review of the Kittler-Young transform method is given in the next section. The use of multilayer perceptron is discussed in section 2.3.2 and section 2.3.3 of chapter 2.

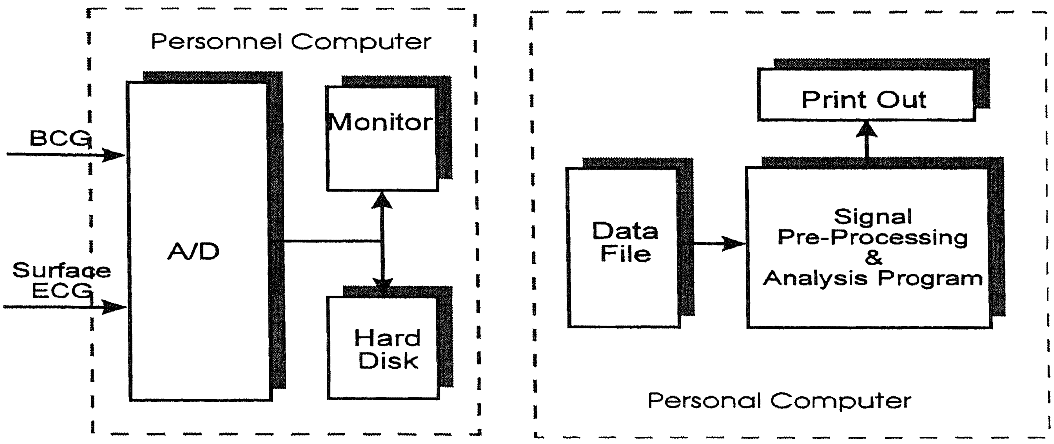


Figure 1.4: A two stage BCG analysis system developed in the University of Cambridge.

In an attempt to develop a home monitoring device, Yamakoshi et al. (1996) reported a technique of recording body weight and ballistocardiogram in the lavatory. A U-shaped weight measuring platform with four strain-gauge type load sensors was designed to measure the body weight and BCG signal while subjects stand on the platform in the toilet. Two channel signals of total body weight and BCG are simultaneously recorded. A microcomputer is used to control the system operation and store data in a 3.5" floppy disk. As indicated by the authors, there are many artefacts in the signals and further improvement of the measurement device is required before conducting discriminate analysis. However, it provides a non-conscious and automated measurement means for health care in daily life.

1.2.3 Computer Assisted Methods for BCG Classification

Some wave complexes of the BCG signals have proven to have directly clinical value. The term 'wave complex' represents local wave amplitude, width and shape within a BCG signal. Brown et al. (1952) have established a qualitative analysis based on the variation in BCG form and amplitude seen during a respiratory cycle, subsequently improved by Starr and Noordergraaf (1967). The records are normally divided into four groups or classes: normal, slightly abnormal, markedly abnormal and extremely abnormal based on the variation in BCG waveform and amplitude during a respiratory cycle. In this classification system, the records which are normal in waveform during the entire respiratory cycle are classified as normal. The slightly abnormal group includes those records in which a minority of complexes during the respiratory cycle are abnormal. In the third group, the majority of complexes are abnormal. In the fourth group, the complexes may be so disorganised that the position of systole cannot be identified with certainty from the BCG alone. However, these methods lack accuracy, especially when the waveforms are contaminated with outside noise, that makes it difficult or impossible to identify either the waves themselves or the position of systole, even for highly trained human observers. Moreover, the visual inspection method is also time consuming.

Talbot (1966) and Harrison (1967) described an automatic method for BCG features analysis. The features rely on the measurement of the wave amplitudes of H, I, and J, slopes of base-line to H, H-J, and I-J, and the wave type such as simple, slur, jog, notch or split, and times of R-H, R-I as well as R-H and R-I in percent of heart period. These parameters are measured by "disinterested operators". The advantage of this technique is that no BCG length adjustment is required. This method was used for classifying two groups of subjects. One group includes 20 normal subjects and the other group are 24 patients with coronary heart disease. By using statistical methods of regression, correlation and discriminate analysis to classify these features, 75% of the correct classification rate was obtained for the whole group.

Due to the time consuming nature of manually obtaining features from BCG waves, several programs to automate or assist with extraction have been developed. Bancroft et. al. (1973) have developed software to automatically identify and to measure the BCG complexes. The operator needs to monitor the curve identification. When a required position is identified, analogue to digital conversion is started by depressing a key. The program first identifies the baseline and then locates the wave peaks according to the baseline crosses. A programme for automatic detection and location of the amplitudes of the static charge-sensitive bed BCG has been implemented (BIOREC Co., 1994). By combining this method and moving cross-correlation techniques, the computer assisted method for the investigation of the relationship of the BCG, respiration variation and the body movements have been developed (Polo, 1992).

Hanka (1979) has proposed a statistical pattern recognition technique using the information contained in the components of the BCG waveform to classify 131 recordings into the normal group and pathological group. The BCG signals were averaged and normalised into standard length of 100 sample points. A distance measure, the Mahalanobis distance, is adopted as an alternative to Euclidean distance for BCG signal clustering analysis. Euclidean distance favours hyperspherically shaped clusters of equal size. The Mahalanobis distance can be used to take care of data sets which do not have spherically shaped clusters or equal size. The squared Mahalanobis distance between pattern vector X_i and the mean of the k th class M_k is defined as:

$$d(X_i, M_k) = (X_i - M_k)^T C_k^{-1} (X_i - M_k) \qquad (1.1)$$

where C_k^{-1} is the inverse of the covariance matrix of the k th class, and k is the number of classes and i is the number of patterns. He found that diseases and BCG wave peaks were related. Large distances show relationships with peaks H and I (Zhao, 1994). The largest ten Mahalanobis distances were used as features for classification analysis. A discriminate classifier was implemented to discriminate the features into two classes. The whole data set was divided into training and testing sets, and the training set was used to estimate the parameters of the discriminate function and the testing data was used to test the classifier performance. A misclassification rate of 4% on the abnormal group was achieved. However, the drawback of Mahalanobis distance is that the calculation of the sample covariance matrix is computationally expensive.

A complicated classification system has been implemented on a computer by Jansen and Shankar (1993). This classifier is based on the assumption that the features have a multi-variant Gaussian probability density function. The BCG signals recorded from the SCSB are digitised at 200 Hz. A number of 30-s intervals are selected for each sleep stage. These data are processed sequentially to extract 15 features which can be divided into five groups. The first group of features are zero-crossing measurements which contain (1) the number of zero crossings of the signal; (2) the number of zero-crossings of the first derivative of the signal; (3) the number of zero-crossings of the second derivative of the signal. The second group of features include three features developed by Hjorth (1970) for use in quantitative electroencephalogram (EEG). The third group are waveform measures which are : (a) the mean and the variance coefficients of the difference in amplitude between successive extremes; (b) the mean and the variance of the time difference between successive extremes; (c) the mean and variance of the time difference between successive inflection points. The fourth group of features is power-spectral in the 3-6 Hz band. The fifth group of features are the heart rate and its variability measurements.

Six volunteers were recorded. Half of the data was used for training and the rest for testing. A classification rate of between 52% and 75% was obtained for sleep staging and the wake state by using a quadratic classifier. These rates improved to between 78% and 89% for classification between awake, rapid eye movement (REM) sleep and non-REM sleep.

Following the early study using statistical pattern recognition techniques conducted by Hanka (1979), Rohen (1985) investigated using nearest-neighbour classification for the whole BCG waveform to classify the normal group, mild hypertension group, coronary artery stenosis group and recent myocardial infarction group. The main contribution of the work is to solve the problem of variation of BCG signal lengths due to the differences between individuals and also between heart beats of the same subject (Rohen, 1985). The proposed algorithm adjusts the individual waves to a standard time window of 80 sample points while retaining all the information contained in the shape of the original signal. It was shown that the nearest-neighbour classifier performs well in discriminating the four groups.

In a recent pilot study using artificial neural networks for BCG signal processing, Zhao (1994) proposed a combined statistical method and feedforward neural network for chair BCG signal classification. Each beat of the BCG signal is scaled within a standard window of 80 points. Altogether, there are 339 samples for the study. Because the available data set is small compared with network free parameters to be determined if the whole BCG waveform is used as input, a statistical feature extraction approach is adopted to reduce the feature dimensionality. The feature extraction is based on the Kittler-Young (K-Y) transform (Kittler and Young, 1972). Generally speaking, the K-Y method is developed from the Karhunen-Loeve (K-L) expansion method. Unlike other K-L based methods, the K-Y method takes into consideration the discriminatory information contained in the class means as well as the class variances. Let input pattern vector $X = [x_1, x_2, \ldots x_N]^T$ belong to one of 3 possible classes ω_i ($i = 1,2,3$), where the probability of occurrence of i th class is $P(\omega_i)$ and ^{T} denotes the transpose operator. The K-L expansion is carried out on the centralised patterns,

$$Z^T = X \cdot U \qquad (1.2)$$

where U is the matrix of the system eigenvectors of the covariance matrix C, which is described in section 2.2.1.

The feature vectors derived from equation 1.2 are transformed by a linear transformation:

$$G^T = Z^T \cdot S \qquad (1.3)$$

where $S = (s_1, s_2, \ldots, s_n)$ is the standard deviation vector of the i th feature (i=1,..,n) and $s_i = 1/\sqrt{\lambda_i}$. λ_i are their associated eigenvalues.

As a result of these two transformations, the centralised input patterns X is transformed into a new feature vector $G = (g_1, g_2, .., g_n)^T$, the K-L expansion procedure is applied again to the entire transformed input training pattern G,

$$F^T = G^T \cdot B \qquad (1.4)$$

where $F = (f_1, f_2, \ldots, f_n)^T$ is the final feature vector and the matrix B is the system of eigenvectors of the covariance matrix $C_g = \sum P(\omega_i) E\{g_i g_i^T\}$. The whole K-Y transform for feature extraction involves three transformation calculations:

$$F^T = X^T \cdot U \cdot S \cdot B \qquad (1.5)$$

The K-Y method was implemented on the BCG records and the three principal features f_1, f_2 and f_3 were extracted from the whole data used for classification by a one hidden layer network. This work provides a foundation of the combination of standard statistical method and neural network for BCG signal analysis. However, there is no test data to test the robustness of the eigenvectors obtained in this study. The features extracted were based on the statistics of the whole data set. Thus, the generalisation of this feature extraction method is still in doubt. Moreover, this method can be computationally somewhat expensive due to repeatedly solving the largest eigenvalues and their corresponding eigenvectors to obtain the matrix U and B. When a new sample X is added, all the computation for solving matrix U, S and B need to be repeated again. This method has the limitation similar to other conventional statistical methods in that it is not suitable for real applications where data comes incrementally (Xu and Yuille, 1995).

1.3 Limitations of Existing Methods

Most existing BCG analysis methods have a disadvantage for real applications in domestic situations in that they rely on two stages; first, recording the BCG waveform and, second, playing back to digital computers to analyse later. The time from measurement to obtaining the result could be measured in terms of hours. Furthermore, most of the analytical approaches rely on expensive software packages running on the digital computers. These analysis packages are computationally intensive and the physical space and power consumption needed are considerable. Thus, these BCG systems are suitable for laboratory or clinical applications. Recently, there is an increasing demand in home health care for robust, low-cost and user-friendly instruments which can automatically interpret biosignal information in real time. Therefore, the goals for the design of the BCG classification instruments for home applications should be small and lightweight, and not require the patient to perform complex operation of the instrument. The classification results should be easy to read by the patients in real time. PC based prognostic/monitoring systems, however, are not user friendly in home applications because users often are not computer-literate.

The BioMatt monitoring system developed by BIOREC Co. has shown the possibility that an instrument can be used at the patients' own home (BIOREC, 1994). A commercial computer is used as an automatic data analysis system. Although nowadays the computers are a solution in data processing and have been frequently used in this type of application, in the real domestic applications, users often are not computer-literate. It is required that the system should be easy to understand and operate. Moreover, the automatic data analysis system only provides 15 parameters processed from the basic signals, which aim at offering medical doctors a clear understanding and diagnostic information. No decision making processes are included in the system. Thus, the BioMatt instrument still has limited use as a compact and easy-to-use heart condition assessment device at the patient's home.

It can be seen from the above survey that extensive progress has been made to develop practical and convenient BCG recording techniques on the various forms of bed and chair apparatus. However, the means of BCG signal analysis have barely been influenced by state-of-the-art signal processing techniques. In the BCG features extraction phase, programs for automatically measuring the individual wave parameters are still human dependent, and could miss some additional diagnostic information because of the noisy nature of the BCG recording. Moreover, further knowledge is still required to decide which combination of these feature parameters are the optimal parameters for the specific diagnosis purpose. Statistical methods for feature extraction have mostly been used for BCG analysis.

However, one practical problem of these methods is that their computation complexity dramatically increases with increasing raw data dimensionality, and they have difficulty in incorporating new data. Another disadvantage is similar to that of the statistical classifier which will be discussed next. In real time applications, it is required that the system works reliably and easily.

Many previous classification methods in literature are based on off-line analysis methods using conventional statistical approaches (Talbot et al., 1966; Harrison et al., 1967; Hanka, 1979; Rohen, 1985; Jansen and Shankar, 1993; Polo, et al., 1992). Statistical classifiers such as linear and quadratic discriminate functions (Fisher, 1936) have been studied for many years. In a typical statistical method, a linear discriminate function is determined by formulating a linear function that produces the maximum degree of separation consisting of two or more measured attributes. Linear regression, however, imposes linearity on the mapping function even though the system to be modelled may not be very linear. Transformations of variables to make the data linear can be made to enable linear regression to be as accurate as desired, but achieving this goal requires special assumptions on the statistical distribution of feature vectors or models (Brouwer, 1995). Statistical classifiers also perform poorly when presented with non-linear relationships (Lovell and Bradley, 1996).

There are other techniques that can cope with non-linear relationships. For example, the perceptron (Rosenblatt, 1958) and the back-propagation neural network (Rumelhart et al., 1986) which adjust the parameters of a discriminate function until it is optimal, or at least satisfactory, according to predefined criteria. When distributions are non-linear and strongly non-Gaussian, neural network classifiers may prove to be more reliable (Smith, 1993). Artificial neural networks based signal processing methods have been shown to have significant robustness in processing complex, degraded, noisy, and unstable signals (Hassoun et al., 1994). The highly parallel neural network architecture makes it attractive for hardware implementation in parallel to solve time critical problems. Recently a specific application study showed that ANNs tend to be superior in terms of speed and memory requirements compared to nearest neighbour methods in pattern recognition problems, while performing comparably to the near-neighbour classifier (Weideman et al., 1995). Furthermore, it is known that a properly trained neural network architecture has good generalising capability, that is, it will give good predictions on unseen examples provided that they satisfy some conditions (Rumelhart et al., 1986B; Lippman, 1989; Hertz et al., 1991).

1.4 The Aims and Scope of This Thesis

Currently, one focus of the development of ANNs has been on the intelligent automatic diagnosis instruments in the medical field. The reason for this is that neural networks have the capability of massive parallelism and learning ability. This property of learning through examples is extremely useful when the problems are difficult to model with conventional methods or the nature of the process is unknown. In applying neural networks to real world problems, it has been found that the networks have good generalisation capability. Furthermore, when a network is trained with noisy data, it is often found to perform as a robust classifier (Holmstrom & Koistnen, 1992; Chakrabarti et al., 1995). For heart disease classification purposes, the classifier can be created off-line with available samples. The trained classifier can be implemented to specify the class to which each case actually belongs during on-line processing.

As demonstrated by a series of recent studies (Bortolan and Willems, 1994; Hu et al., 1994; Stamkopoulos et al., 1992), artificial neural networks have provided satisfactory results in the ECG signal analysis field. However, except for Zhao's work using artificial neural networks in BCG analysis (Zhao, 1994), past research has shown very few applications using artificial neural networks for BCG signal processing. Most of the published works are based on traditional statistical analysis and heuristic methods. Therefore, the issues of improving BCG signal processing techniques and computer-aided BCG analysis, such as state-of-art pattern recognition techniques and neural computing, need to be better addressed (Zhao, 1994). The present work will concern these issues in the engineering field.

In the previous work of BCG classification using neural network, Zhao (1994) has addressed the point that if the available data set is small compared to the network size, reducing the feature dimensionality can improve the classification performance. In real applications, a large neural network has the limitations that it requires a long training time and a large circuit to implement it. Conventional statistical methods, such as principal component analysis (PCA), Karhunen-Loeve expansion or Kittler-Young transform are based on computing the input data covariance matrix and then applying a numerical procedure to extract the eigenvectors and the corresponding eigenvalues. However, for large data sets, the dimensions of the covariance matrix grow significantly large making its computation and manipulation practically inefficient. Recently, the development of neural network based PCA shows the ability to extract principal components directly from the input data set. Neural networks have the property of parallelism, speed and trainability. They provide good, if not the best, architectures for fast and efficient processing of large amounts of data and can be adopted for real time application. Using neural networks based PCA as an alternative technique for BCG feature extraction needs to be explored as it is the computationally efficient method in that it does not need to form the data covariance matrix.

The last few years have seen an explosion of multiresolution techniques in signal and image processing research. Multiresolution offers an effective framework for extracting information from signals and images at various scales. Mallat (1989) has shown that the multiresolution wavelet can be computed with a pyramid algorithm based on convolution with quadrature mirror filters. In this way, the computational complexity of signal processing operations is significantly reduced. At each pyramidal process, the signal is decomposed into its coarse and detailed components at various scales. Signals at various resolutions give information about different features. In the classification phase, not every resolution is equally important to provide the classification information. Therefore, choosing the proper resolution and discarding the unnecessary components can be used as an alternative method for BCG dimensionality reduction.

With the advances in computer and electronic techniques, there is an increasing trend in using computer based medical systems to collect, store, and process digitised biological signals. As new technologies and processing algorithms are under development, more computational resources are needed. For a real time system, an instrument may have to deal with a high volume of data in a short time. Traditional techniques using computers (Von Neumann machines) have the problem that the sequential computational hardware may not meet the constraints of the computing time. Thus, it is necessary to consider a particular technique that can efficiently map these algorithms to special purpose hardware to balance these requirements. On the other hand, in many portable medical instrument applications, it is required that the system can quickly process real time information without complex programming. This means that, the system should be able to indicate the class of a vector, representing both input and output as an electrical signal in a short time. The primary requirements in hardware implementation is maximum performance at minimum power consumption and cost. This requirement presents challenges in making decisions in algorithm, circuit technology and design methodology. Although a special technology can be used to enhance the performance and design, the implementation of a new technology is usually very expensive. In this case, commercially available integrated circuits are the cost effective solution. On the other hand, in the system prototype stage, it is desirable that the custom design allows the optimisation of the parameters or new algorithms. Therefore, the hardware platform should be flexible.

It follows from the above discussion that the BCG has the potential to serve as a low cost, non-invasive, easy-to-use, domestic screening procedure. The existing BCG systems are suitable only for laboratory or clinical applications. They have limited usefulness for domestic applications. A convenient, easy to use, small size and inexpensive real time system, which can provide an automatic diagnosis is highly desirable in order to bridge the gap between laboratory research and domestic applications.

The principal aim of the work presented here is to suggest, and to demonstrate, a general framework for developing a practical real time system in BCG signal analysis. The goal is to achieve low cost, small size and ease of use by incorporating a neural classifier into an intelligent portable system. The particular application discussed in this work is to use the BCG to identify patients with a risk of heart attack or with mild hypertension. Furthermore, motivated by the potential requirements of parallel processing in the biomedical field, another goal of this work is to exploit a method for implementing a fully parallel neural network in hardware with off-line training and on-line implementation. It takes up the challenge to explore the trade-off between calculation precision requirement and classifier performance. The implementation of the neural network on a flexible platform, using digital field programmable gate arrays (FPGAs), is explored to demonstrate the programmable technology that enables rapid hardware design with improved performance and the flexibility to meet the research requirements of a variety of biomedical applications. To facilitate acceptance for practical application, the same BCG data are used as in an earlier study by Zhao (1994). These were recorded from the chair in routine practice by the Medical Informatics Unit of Cambridge University.

1.5 Organisation of the Thesis

This thesis has 6 more chapters, which are arranged in the following way. In chapter 2, a method using a neural network to implement principal component analysis for feature extraction is described. Following a brief review of static neural classifiers, the multilayer perceptron (MLP) with error backpropagation training algorithm is adopted in the rest of this study. By using the technology introduced above, a combined PCA network and MLP system are evaluated with both leave-one-out and data partitioning schemes. The performances of linear PCA and non-linear PCA networks for BCG feature extraction are compared. The results are compared with the result derived by Zhao (1994) using the combined Kittler-Young transform method with MLP. Furthermore, the robustness of the combined classification systems to the effect of noise was tested. By analysing the experimental results, it is suggested that the statistical method for feature extraction is not an optimal method, if a class data set is too small compared with other classes. The constraint of the limited data set in this study needs the development of an improved method for feature dimensionality reduction.

In chapter 3, following a brief review of discrete wavelet transforms, a wavelet transform is introduced for time domain BCG dimensionality reduction. The idea is to decompose the original signal into a series of views at different scales, and the coarse components which contain the significant information are presented to an MLP classifier. The performance and computation efficiency of the combined wavelet transform and MLP approach is compared with the results presented in chapter 2. Based on the comparison results, an automatic BCG classification system using combined wavelet transform and MLP is proposed.

In chapter 4, related off-line methods for reducing the weight precision are reviewed. It is shown that when the weight precision becomes lower, the updating change in weight could vanish and the network could not carry on learning. A modified new hybrid learning algorithm is proposed intending to overcome this problem. The proposed learning algorithm combines the conventional backpropagation method and random weight optimisation method to prevent the problem of network paralysis during low precision training. Two benchmark problems are used to test the hybrid algorithm and the results show that the proposed algorithm is superior to the conventional backpropagation learning method. By using the proposed algorithm, it is shown that a 6-bit weight precision MLP can still give satisfactory performance compared with high precision. Therefore, the hardware implementation complexity can be greatly reduced.

In chapter 5, implementing a fully parallel neural network on Xilinx's XC4000 series field programmable gate arrays for specific application is described. The work can be considered as extension of the work of Cox and Blanz (1992) and Botros and Abdul-Aziz (1993). However, instead of complex adder design to improve performance, a simple ripple carry adder tree is adopted to make use of the fast carry logic feature of the XC4000 series to achieve the performance. The sigmoid function is implemented with lookup tables. In chapter 6, the method of mapping each hidden neuron with 20 inputs to one FPGA and 3 output neurons to one FPGA is demonstrated for the Xilinx XC4000 series. The design is based on the combined VHSIC hardware Description Language (VHSIC is short for "Very High Speed Integrated Circuit") and schematic entry method which make the later parameters update and network expansion more flexible. The work uses available resources for design, synthesis and timing analysis of hidden and output neurons. Resources were not available to transfer the design into hardware. The neurons implementation and time analysis results suggest that the proposed design has a maximum 37 nano-second delay which is possible to give 27 million decisions per second performance for a physical hardware system.

Finally, Chapter 7 of the thesis provides conclusions of this study. The methodology presented in this thesis is the first step towards developing on-line BCG classification models for portable instrument applications. A model which combines wavelet transformation and multilayer perceptron is developed for on-line BCG classification. A method for implementing an automatic BCG classification instrument is presented. A new hybrid learning algorithm is developed for weight precision selection of the neural classifier to reduce the hardware implementation complexity. Furthermore, the approach of off-line training and on-line implementation of a feedforward neural network with FPGAs has been demonstrated, that has various potential applications in the biomedical signal and image processing field. Finally, a discussion of future work is given.

Chapter 2

Combining a Principal Component Analysis Neural Network and a Multilayer Perceptron for BCG Classification

2.1 Introduction

In chapter 1, it is indicated that most of the existing BCG classification methods use off-line processing techniques; that is, previously recorded BCG patterns are analysed in the laboratory. It is desirable to develop an on-line classification system which can operate quickly in situ. In the on-line method, the classification system needs to connect to the BCG sensors and to automatically interpret the information in real time. Among the advantages of automated processing are objectivity, repeatability, and speed. The objective of this chapter is to investigate an on-line BCG classification system based on a combination of a neural network that performs principal component analysis and a multilayer perceptron classifier. The term "combined neural network" is used for this system in this work.

Biological inspired neural networks have several properties that make them promising in real world applications. They are capable of learning patterns of an arbitrary nature by using simple general learning algorithms. Once trained with data which is sufficiently representative, the neural network can perform correctly on inputs that it has not seen during the training and produce the desired output. Neural networks also exhibit very good performance on noisy patterns which they previously learned (Hassoun et al., 1994). These properties have made them widely used for automatic signal processing problems. For example, Widrow and Winter (1988) have implemented an adaptive filter on a neural network for fetal ECG diagonal analysis. Chakrabarti et al. (1995) illustrated that a multilayer feedforward network provided a higher successful classification rate than a conventional minimum distance classifier for radar target classification. In the last few years, several versions of feedforward neural classifiers have been applied in the field of medical classification and diagnosis. In a diagnostic ECG classification experiment, Borrblon and Willems (1994) indicated that a feedforward neural network trained with backpropagation rule has better accuracy than conventional statistical methods such as classical linear discriminate analysis in the problem of diagnostic classification of the electrocardiogram. Farrugia et al. (1993) illustrated a MLP approach to the classification of arrhythmia from the ventricular intracardiac electrogram. A feature extraction and neural classifier scheme is adopted. The experimental results show that the artificial neural network performed better than a rate-based scheme used in some commercial devices. Overall, these reported works indicate that neural network technology provides a much better performance than the conventional statistical classifier in solving medical pattern classification problems. The BCG patterns classification has a similarity with ECG signal analysis by considering amplitudes and timing of different peaks. Thus, the capability of neural network in ECG signal analysis suggests that the artificial neural networks provides an attractive alternative for BCG signal classification.

In a previous work, Zhao (1994) has illustrated a framework using a single feedforward neural network to classify whole time domain BCG waveforms. The input to the neural network is a standardised BCG wave pattern with 80 sample points. The results showed that with the available data of 339 samples, the misclassification rate for heart attack subject was 66.7%, which is not acceptable in prognostic applications. One obvious problem is that the error was caused by the low ratio of heart attack class samples to the feature dimensionality in the training set, which is 0.075 (Zhao, 1994). In a theoretical framework, Baum and Haussler (1989) suggested that the number of samples needed to achieve a generalisation error rate of ε is roughly the number of weights divided by ε for classification problems. Therefore, it is necessary to reduce the input dimensions to minimise the network size for such limited samples. Furthermore, if a high sample rate is implemented in some BCG systems, a classification of the whole BCG waveform would be computationally intensive and time wasting by the artificial neural networks, because the high dimension input forces us to use one large network architecture with a high number of free or effective parameters (i.e. the number of adaptive weights and neurons). Moreover, the presence of irrelevant feature combinations can also obscure the impact of the more discriminating inputs. There is considerable empirical evidence that networks can be made to perform well on a given task by minimising the size of the network and reducing number of weights (Hammamoto et. al., 1996). Thus, if the number of inputs to the ANN can be reduced without significant loss of information, then it should be possible to successfully classify the signal based on a reduced or compressed data set. There are several advantages of using feature extraction in the pattern classification field. First, reducing the dimensions of the data makes the problem more over-determined by the training set and therefore can increase the classification rate; second, reducing the dimensions of the data speeds up the training; and last, features can incorporate invariance such as scale, translation, etc., to avoid impractically large training sets. Finally, in real time applications, it is important to reduce the number of features to cut down the information storage and computation cost, not only in software but also in hardware as well.

Zhao (1994) introduces a statistical method for feature extraction and shows that using extracted features as inputs to a neural classifier provides a better performance than a single neural classifier approach. The presented work was based on a statistical method of K-Y transformation (Kittler & Young, 1979) for BCG feature extraction. As stated in Chapter 1, the K-Y transform requires expensive computation power to extract the eigenvalues and corresponding eigenvectors which make it inefficient to adapt to new data in real time applications. The proposed K-Y transform is a modification of the Karhunen-Loeve (K-L) expansion. Principal component analysis (PCA) and the closely related K-L expansion are well established statistical methods in many engineering and scientific feature extraction and data compression applications (Devijver and Kittler, 1982). It is an optimal method for determining the minimum number of linear combinations of the input data that permit accurate reconstruction of the data. In recent years, neural network approaches to principal component analysis using Hebbian learning (Oja, 1982) have attracted a tremendous interest. They can learn directly from input patterns to obtain expected solutions. Only a simple learning algorithm associated with simple computations is required, so that, they can be implemented for on-line data processing (Mao and Jain, 1995). The neural extension approach for PCA has received much attention in the domain of feature extraction, especially within signal processing, data compression, and automatic target recognition applications (Sanger, 1989; Kung, 1993; Nagasaka and Iwata, 1993; Rogers et al., 1995; Ramanan, et al., 1995). It has been shown that PCA networks have good generalisation capability in image processing (Bannour and Sadjadi, 1995). Thus, this capability of PCA networks can be used to extract features if the data has consistent statistical properties. Moreover, their architectures are also suitable for hardware implementation with very large scale integrated (VLSI) circuits for real time application (Kotilainen, et al., 1992).

In view of Zhao's earlier work (1994), the objective of this Chapter is to explore a computationally competitive method based on PCA networks as an alternative method for on-line BCG feature extraction. In section 2, both linear and non-linear training rules for neural network implementation of PCA are described. In section 3, following a brief review of several neural classifiers, a multilayer perceptron model with backpropagation training is adopted for experimentation. Section 4 evaluates the combined PCA network and MLP approach in BCG classification. Based on simulation results, a comparative analysis of the proposed approach and the method describe by Zhao (1994) is presented in section 5, and section 6 concludes the chapter.

2.2 Principal Component Analysis Neural Networks for Feature Extraction

2.2.1 Principal Component Analysis

Principal Component Analysis exploits the statistical regularities of input data, extracting and compressing the information contained in it. The compression is achieved by projections of the input data from their n-dimensional space onto an m-dimensional output space ($m << n$) which retains as much as possible of variation of the input data. This reduction is achieved by projections of the data samples onto the eigenvectors of the data correlation matrix. The principal components (PC) thus obtained are ordered so that the first few retain most the variation present in the data. For the detailed mathematical theory of PCA, please refer to Joliffe (1986).

Principal components are usually derived by numerically computing the eigenvector of the variance matrix. Those eigenvectors with largest eigenvalue (variance) are used as features onto which the data are projected (Joliffe, 1986). This method may be considered optimal, in the sense that mean square linear reconstruction error is minimised. For a given data set $X = (X_1, X_2, \ldots X_N)$ with a zero mean, a simple approach to PCA is to compute the simple variance matrix

$$C = \frac{1}{N}\sum_{i=1}^{N} X_i X_i^T \qquad (2.1)$$

The next step is to use any of the existing standard eigenvector analysing algorithms to find the m largest eigenvalues and their corresponding eigenvectors of the data covariance matrix:

$$C\phi = \lambda\phi \qquad (2.2)$$

where λ is the eigenvalue, and ϕ is its corresponding eigenvector. The m principal components of n dimensional data are the m orthogonal directions in n spaces which capture the greatest variation in the data. These are given by the eigenvectors ϕ .

This standard eigenvector-analysing-based approach is expensive in computation. So the approach is not suitable for real applications where data arrives incrementally or on-line (Xu & Yuille, 1995).

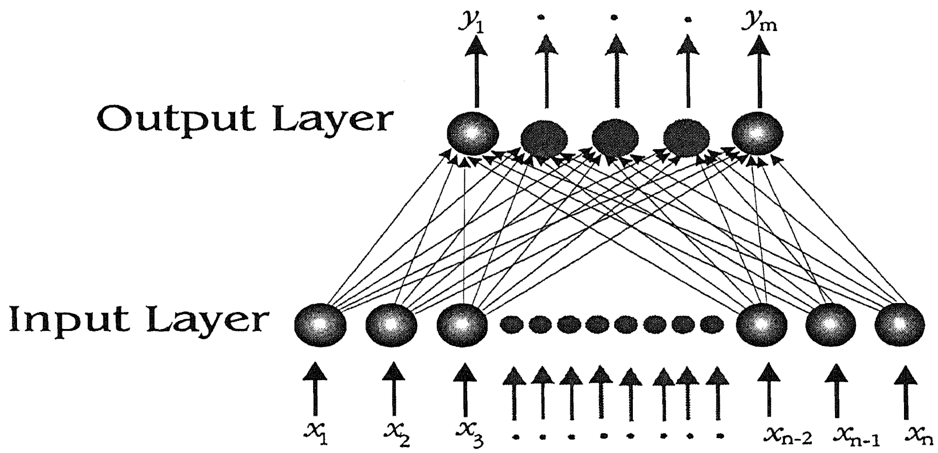


Figure 2.1: PCA network architecture

2.2.2 Principal Component Neural Networks

A variety of two layer neural network learning algorithms for extracting multiple principal components have been proposed (Sanger, 1989; Rubner and Tavan, 1989; Oja, 1992; Xu, 1993; Matsuoka and Kawamoto, 1994; Palmieri and Zhu, 1995). Most of them are based on Oja's earliest one-unit algorithm (Oja, 1982) or similar, though not identical. From the hardware implementation point of view, Oja's modified subspace rule is chosen in this study because of its properties of homogeneity and locality, and because it does not require any externally "hardwired" asymmetry such as the model proposed by Sanger

(1989), nor lateral orthogonalization networks as proposed by Rubner and Tavan (1989). More recently, the PCA networks, which stemmed from Oja's constrained Hebbian rule, have been extended from linear cases to non linear cases (Oja, 1992; Baldi and Hornik, 1995; Karhunen and Joutsensalo, 1995; Xu, 1993; Palmieri and Zhu, 1995). The basic two layer PCA network is shown in Figure 2.1. The architecture consists of an input layer and an output layer. The learning rule extracts a weight matrix that spans the same subspace as the m largest principal components.

2.2.2.1 Linear PCA Network

Oja has shown that a single linear neuron with a Hebbian learning rule is able to extract the largest principle component from its training inputs by forcing the weights to have a constant value (Oja, 1982). Let the $X = [x_1, x_2, \ldots x_n]$ denote the n-dimensional input vector. For a one output network, the output is $Y = WX^T$. The purpose of the learning algorithm is to find a suitable weight vector $W = (w_1, w_2, \ldots w_n)$. In other words, W is a principal component and Y is the score on the principal component. The one-unit Oja's learning rule at t th iteration can be written as follows:

$$\Delta W(t) = \eta(t)Y(t)(X(t) - Y(t)W(t)) \qquad (2.3)$$

where $\eta(t) > 0$ is the learning rate parameter. $\Delta W(t)$ is the change of $W(t)$ on presentation of the t th input data vector. This rule is used to train a linear neuron whose output $Y(t)$ is equal to the product of weight vector $W(t)$ and input pattern $X(t)$ at time t.

Based on the single linear processing unit PCA model, Oja extended the learning rule to m linear processing units. The activation of each neuron is a linear function of its inputs. The architecture of the network is shown in figure 2.1. Suppose the output of j th neuron is $Y_j = W_j X^T$, where $j = 1,2,\ldots m$. The weight vector update rule is expressed as:

$$\Delta W_j(t) = \eta(t)[X(t)Y_j(t) - Y_j(t)\sum_{i=1}^{m} W_i(t)Y_i(t)] \qquad (2.4)$$

This algorithm is called the symmetrical subspace learning algorithm. It learns the space spanned by the eigenvectors corresponding to the largest m eigenvalues of the data correlation matrix, but usually not the true eigenvectors themselves. Therefore, it does not generate true principal components. Recently, a new modified subspace algorithm that can perform true PCA has been proposed by Oja et al. (1992), and independently proved by Xu (1993). This property is also confirmed by Palmieri and Zhu (1995). The modified learning rule enables the symmetrical network to perform PCA by introducing a scalar parameter to each weight vector. The modified learning rule can be expressed as:

$$\Delta W_j(t) = \eta(t)[X(t)Y_j(t) - g_j Y_j(t)\sum_{i=1}^{m} W_i(t)Y_i(t)] \qquad (2.5)$$

where $G = [g_1, g_2, \ldots g_m]$ are arbitrary positive parameters with $g_1 > g_2 > \cdots > g_m$. It is proved that the parameters g_j break the symmetry in the subspace learning rule and ensure that the weight vectors will tend to the true PCA eigenvectors.

2.2.2.2 Non-Linear PCA Network

By introducing non-linearity units into the output units in the PCA network, the Hebbian learning rule can be extended to non-linear PCA networks (Xu, 1993; Oja et al., 1992; Karhunen and Joutsensalo, 1995). For the case of the Hebbian learning rule in a linear unit, the objective function projects the input patterns in the directions of maximum variance. The same criterion function applied to non-linear units, on the other hand, may lead to results radically different from those produced by linear units. Both linear and non-linear learning rules are seeking a set of weight parameters such that the outputs of the unit have the largest variance. The non-linear unit, however, constrains the outputs z to remain within a bounded range, e.g., $Z = \tanh(\beta Y)$ limits the output within [-1, 1]. Oja and his co-workers (1991) proposed various possible non-linear activation functions and their corresponding criterion functions. Karhunen and Joutsensalo (1993) showed that the utilisation of units with non-linear activation functions which employ Hebbian learning may lead to robust principal component analysis. Sudjianto and Hassoun (1995) demonstrated that non-linear Hebbian units have better capability in data clustering. Thus, it has potential applications in feature extraction. The non-linear Hebbian learning rule used in their study is one of Oja's proposed variations as shown in equation 2.4. The learning rule is as follows (Sudjianto and Hassoun, 1995),

$$\Delta W_j(t) = \eta(t) Z_j \frac{dZ_j(t)}{dY_j(t)} [X(t) - \sum_{i=1}^{m} W_i(t) Y_i(t)] \tag{2.6}$$

2.2.2.3 Summary of PCA Network Training Algorithms

The above linear principal component analysis proposed by Oja et al. (1992) and Xu (1993), and non-linear principal component analysis proposed by Sudjianto and Hassoun (1995) are used in this study for BCG feature extraction. These algorithms can be summarised as follow:

1. Initialise all connection weights to small random values and choose the values of the learning parameters.
2. Randomly select an 80-dimensional BCG pattern and present it to the network.
3. Update the connection weights between the input nodes and the output layer using equation (2.5) for the linear PCA network and equation (2.6) for the non-linear PCA network respectively.
4. Modify the learning rate parameter.
5. If the iteration value reaches the defined criteria, then stop; else go to step 2.

2.3 Neural Network Classifiers

Today, artificial neural networks comprises a well established field of research. As the term 'neural network' implies, it was originally aimed more towards modelling networks of real neurons in the brain. The subject reached a first peak in the 1940's due to McCulloch and Pitt's work (McCulloch and Pitts, 1949), then again in the 1960's with Rosenblatt's perceptron convergence theorem (Rosenblatt, 1962) and the very similar networks called ADALINES invented around the same time by Widrow (Widrow, 1962). Rosenblatt's main objective was to demonstrate analytically and experimentally that adaptive neural networks with a rich interconnectivity and synapse could mimic many observed cognitive functions, and that the existence of such structures did not conflict with the available biological evidence. However, the limitation of Rosenblatt's classifier in its inability to solve problems which are not linearly separable, was discussed by Minsky and Papert (1969) in their book "Perceptrons". In their book, Minsky and Papert suggested that it would be more profitable to explore other approaches to artificial intelligence. Although there were still a number of people who continued to develop neural network theory in the 1970's (Anderson, 1970; Marr, 1969; Kohonen, 1974; Grossberg, 1969; etc.), attention left the neural network paradigm for nearly 20 years. It returned finally for a third time in the 1980's with the Hopfield energy approach in 1982 (Hopfield, 1982) and the back-propagation learning algorithm for multilayered feedforward networks which was proposed first by Werbos (1974) in the mid-70's, and then independently rediscovered around 1985 by Rumelhart and his PDP project colleagues (Rumelhart et al., 1986B).

2.3.1 Related Neural Based Classifier Techniques

The attractive property of neural networks for real world problems is that, after a number of presentations of the entire training samples, the networks can self stabilise, and can read out the expected output for each input without search. This property has been used in the field of classification and has led to the development of various "neural" classifiers. In relation to the neural networks learning algorithms, the neural classifiers can be categorised into two main paradigms: supervised networks and unsupervised networks. In the supervised group, these include the multilayer perceptron (MLP), radial basis functions (RBF), decision based neural networks (Kung, 1993), Boltzman networks, and supervised learning vector quantization (LVQ) etc. The unsupervised group of neural-like schemes such as the adaptive resonance theory (ART) map, self organising map (SOM), principal component analysis networks which have been described in the previous section and Hopfield network etc., have also received considerable attention.

The strength of both groups of classifiers lies in their applicability to problems involving arbitrary distributions. For an unsupervised learning rule, no external teaching is required. The network learns to form internal representations or codes from the input data, or correlation between patterns, and organises patterns into categories from these correlations. SOM and ART networks have been used in a number of applications including robotics control, speech and pattern recognition. However, the SOM network usually needs more neurons to attain a fixed level performance, and the general architecture of ART networks are more complicated. Both models are relatively inefficient in their use of output neurons. The properties of PCA networks are mostly applied to the feature extraction and data clustering applications. This study focuses on supervised learning models for classification.

Supervised learning networks dominate the mainstream in pattern recognition applications. Usually the training data consists of many pairs of input/output training patterns. Adaptation occurs when the system directly compares the network output with a known correct or desired answer. Therefore, the learning will require the assistance of teachers. Integrating neural networks with other soft-computing tools such as fuzzy logic and evolutionary computations also provides a much more attractive and strong computational methodology for solving complex pattern recognition problems.

2.3.2 Static Classifier

There are two types of classifiers in the supervised learning group. One is the static classifier such as the MLP, RBF, and LVQ classifiers mentioned above and the other is the recurrent neural network for spatio-temporal information processing such as Hopfield networks and dynamic recurrent networks. Although these dynamic classifiers are more powerful and promising for complex computations than static networks, the added complexity to the network means that they need longer training schedules and increasing computational power. To favour simplicity for real time implementation, the considerations in this study are restricted to the static classifier only.

One of the advantages of the static classifier is its generalisation ability, which means that after the network has been properly trained, it has the ability to classify unseen data. This generates a wide interest for real time applications. Besides the mostly populated MLP, RBF, and LVQ type static classifiers, there are several other neural-like candidates such as polynomial networks including higher order networks, sigma-pi networks and some functional link architectures (Shin and Ghosh, 1995).

Multilayer feedforward neural networks that adapt weights using gradient descent of the mean squared error (MSE) in weight space, are the most commonly used neural network classifiers. The most popular kinds of multilayer feedforward networks are multilayer perceptrons in which each computational unit employs a non-linear transfer function which mostly is a sigmoidal function. Hornik et al. (1989) proved that the MLP is capable of approximating any function of interest. The development of the backpropagation learning algorithm for determining weights in a multilayer perceptron has made these networks the most popular among researchers and users of neural networks. The MLP classifiers have good performance with noisy signals. MLP methods are simple and need less memory compared to the other classifiers, which makes them good candidates for hardware implementation.

Radial basis function networks belong to the group of kernel function networks that utilise simple kernel functions, distributed in different neighbourhoods of the input space. Each hidden node in the RBF network represents one of the kernel functions. . The main difference between MLP and RBF networks lies in the fact that basis functions used in the hidden-layer nodes of RBF networks are Gaussian, while the MLP uses sigmoid functions. It has been shown that RBF networks with one hidden layer can also be used as universal approximates (Hartman, et al., 1990). Applications of RBF networks as classifiers have been reported in the literature. Roy et al. (1995) presented extensive comparison experiments between RBF and other techniques for a wide variety of classification problems. The computational results show that the RBF classifier works quite well on those problems. The results are robust with respect to variations in the class distributions. However, the RBF networks need much shorter training times at the expense of additional memory as compared to the MLP. Moreover, for many problems, the RBF network often involves a large number of hidden units. This results in that the run time (after training) speed of the RBF networks is often slower than that of a MLP.

The LVQ is a nearest neighbour classification method which assigns an unclassified sample vector to the nearest class of a set of correctly classified prototypes, or codebook vectors (Kohonen, 1989). It has been reported that the further refinements such as LVQ2, and LVQ3 need somewhat less training time than an MLP-based classifier for comparable performance (Geva and Sitte, 1991). The memory requirements are also similar. However, experimental results show that their performance is more sensitive to the initial choice of reference vectors (Song and Lee, 1990).

Static classifiers offer such a wide variety choice that it is not practical to construct exhaustive computation experiments to compare them all for a single problem. Furthermore, Ng and Lippmann (1990) show that classification error rates are similar across different classifiers when they are powerful enough to form minimum error decision regions, when they are properly tuned, and when sufficient training data is available. Practical characteristics such as training time, classification time, and memory requirements, however, can differ by orders of magnitude. The feasibility of hardware implementation has also constrained the neural model selection. Taking these factors to account, the multilayer perceptron is adopted as the BCG classifier for this study due to its popularity and its ability to learn non-linear functions. The multilayer perceptron networks have found numerous applications in pattern recognition and function approximation (Hertz et al., 1991). They have been successfully used in a wide range of medical applications (Farrugia et al. 1993; Hu et al., 1994; Dassen et. al., 1994; Nekovei and Sun, 1995; Leong and Jabri, 1995). They also show a good fault tolerance to noisy data (Reed et al., 1995).

Although there are some existing theoretical methods in literature providing guidelines for considering the design issues such as how many layers are needed for a given task, how many units are needed in the hidden layer and how large the training set should be for "good" generalisation, for many task specific applications, many design parameters still must be determined by trial and error (Bortolan and Willems, 1994; Hu et al., 1994; Nekovei and Sun, 1995). This is because these guidelines are derived under the assumption that the training data is ideal and sufficiently large. In real world applications, the data samples may be limited and also noisy. Therefore, in this study, a three-layer feedforward neural network is adopted and the number of hidden nodes is decided empirically. The architecture of the three-layer feedforward neural network is shown in Figure 2.2. The hidden layer and the output layer are connected to bias elements.

2.3.3 MLPs Learning Algorithm

Back-propagation is a popular training algorithm for MLPs, and has been widely recognised as a powerful tool for learning input-output mapping in many neural network applications. However, it has several well known drawbacks, which are the slowness of the learning process, especially when large training sets, or large networks have to be used; the lack of good theoretical results to determine the optimal network architecture for a given task; and also there is no guarantee that the network will converge to the global minimum error. It is a rich and open research area to improve the network learning efficiency. The speeds of algorithms have been significantly improved during the last few years by introducing many different approaches. For example, it has been shown that the learning speed can be improved by dynamically adapting the learning rate (Jacobs, 1988; Tollencere, 1989), or by using new error functions (Humpert, 1994), or by using second order information (Battiti and Masulli, 1990; Becher and Le Cun, 1988). Methods for the optimisation of network architecture have also been developed, such as pruning the network to remove the non-useful neurons and connections during training (Sietsma and Dow, 1988; Giles & Omlin, 1994), or including prior task specific knowledge in the network initial design (Courrieu, 1993). To overcome the problem of the network stuck in a local minimum, possible modifications including simulated annealing (Masters, 1993) and applying genetic search technique (Lee, 1996) have been proposed. Researchers have shown that proper initialisation of weights can alleviate the local minima problem (Denoeux and Lengelle, 1993). More recently, Yu and Chen (1995) found that the error surface of a backpropagation network with one hidden layer and $N-1$ hidden sigmoid units has no local minima, if the network is capable of implementing an arbitrary training set consisting of N different patterns.

Although there are many improvements on network learning algorithms, no study has shown a clear advantage of one method over others in convergence ability, architecture optimisation, generalisation capability and learning speed. The good performance of a method in one application does not assure that it will be the same for other problems which have different emphasis in difficulties (Finnoff et al., 1993). Moreover, most of these methods solve individual problems by introducing a number of extra independent free parameters which causes extra cost of computational power. Because in many real applications a network needs only one training, and then it may be used for a long time, the difficulty of long learning time may not affect the network performance. In exhaustive comparison experiments using a wide range of modifications to the backpropagation algorithm to train a one hidden layer neural network on a real-world medical signal classification problem, Alpsan et al. (1995) have concluded that for pattern classification tasks, when the training set is learned within a certain error tolerance, the standard backpropagation rule may well be good enough compared with more sophisticated second order methods. Thus, considering its desirable properties as a classifier, including its success at generalisation and its complex decision boundary, the MLP with back propagation learning was used as the neural classifier in this study for BCG signal classification.

2.3.3.1 Standard Backpropagation Rule

The Back-propagation (BP) method described by Rumelhart and his co-workers (1986A) has been widely used to adapt artificial neural networks for various actual pattern classification problems. This standardised learning rule is briefly described in the following for a three layer feedforward neural network. As shown in Figure 2.2, the input layer consists of N nodes, the hidden layer has K nodes, and the output layer contains M nodes. A bias neuron θ is employed in the input and hidden layer. For a given p th input pattern of $x_{p1}, x_{p2} \ldots x_{pN}$,, the network produces output o_{p1} to o_{pM}, at the output layer. The total difference between ideal output and real output of the neural network is defined by the following cost function:

$$E = \frac{1}{2}\sum_{p} (o_{pm} - d_{pm})^2 \qquad (2.7)$$

where o_{pm} and d_{pm} are the actual output from the m th unit of the output layer corresponding to the p th pattern and the ideal output shown by the teacher, respectively. Let w'_{nk} denote the weight connection between the n-th input node (where n=1,2,..., N) and the k-th hidden node (where k=1,2,...,K). The output of the k-th node in the hidden layer for the p-th pattern is given by:

$$o'_{pk} = f(\sum_{n=1}^{N} w'_{nk} x_{pn} + \theta) \qquad (2.8)$$

Similarly, let w_{km} denote the weight connection between the k-th hidden node and the m-th output node (where m=1,2,...,M). the output of the m-th node in the output layer is given by:

$$o_{pm} = f(\sum_{k=1}^{K} w_{km} o'_{pk} + \theta) \qquad (2.9)$$

where the f is the logistic sigmoidal function used in this study which is defined as follows:

$$f(output) = 1 / (1 + e^{-output}) \qquad (2.10)$$

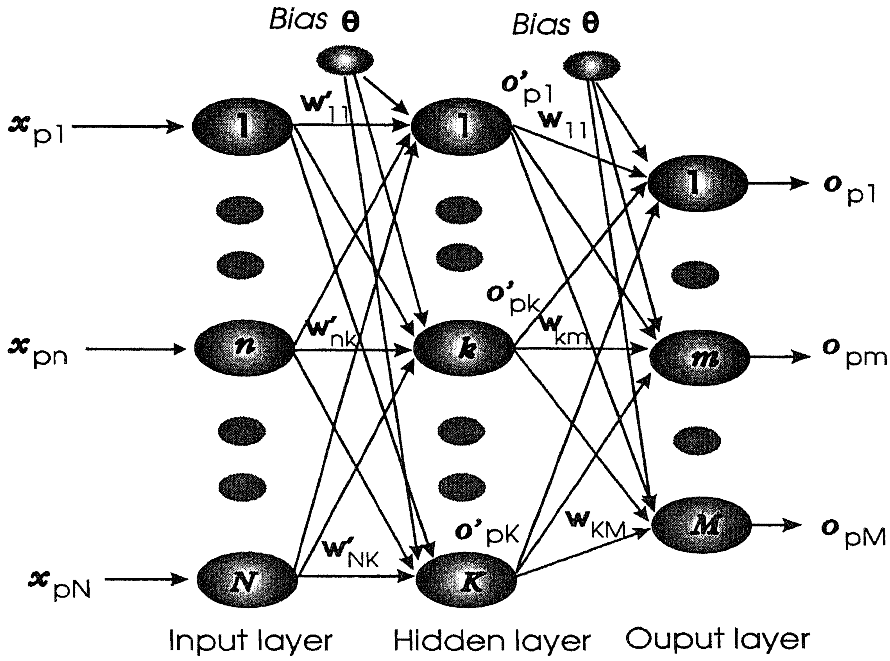


Figure 2.2: The architecture of a neural network with one hidden layer.

The standard BP learning algorithm involves finding a set of network weights that minimises the cost function E for the given training set. An initial set of weights is randomly chosen, and then the weight update starts from the output layer and proceeds backward layer by layer. For the pth input pattern at the (t+1)th iteration, the weight connection between hidden nodes and the output nodes can be expressed as follows,

$$\Delta w_{km}(t+1) = \eta \sum_{p=1}^{P} \delta_{pm}(t) o'_{pk}(t) \qquad (2.11)$$

where the δ is the error signal.

$$\delta_{pm} = (d_{pm} - o_{pm}) o_{pm} (1 - o_{pm}) \qquad (2.12)$$

The weights changes between the input nodes to the hidden nodes can be expressed as:

$$\Delta w'_{nk}(t+1) = \eta \sum \bar{\delta}_{pk}(t) x_{pn}(t) \qquad (2.13)$$

where

$$\bar{\delta}_{pk} = o_{pm}(1 - o_{pm}) \sum_{m=1}^{M} \delta_{pm} w_{km} \qquad (2.14)$$

It has been shown in the literature that an extra momentum term, γ, can be used to accelerate the convergence time and to prevent getting suck at a local minima (Rumelhart et al., 1996). In this case, the weight update equation (2.11) and (2.13) can be modified to

$$\Delta w_{km}(t+1) = \eta \sum_{p=1}^{P} \delta_{pm} o'_{pk} + \alpha \Delta w_{km}(t) \qquad (2.15)$$

$$\Delta w'_{nk}(t+1) = \eta \sum_{p=1}^{P} \bar{\delta}_{pk} x_{pn} + \alpha \Delta w'_{nk}(t) \qquad (2.16)$$

This weight update rule is used in the rest of this chapter and in chapter 3 for network training.

2.4 The Combined Neural Networks Approach for BCG Classification

2.4.1 Description of BCG Data

The data used in this study were obtained from the Medical Informatics Unit of the University of Cambridge derived from a field trial to assess the suitability of a BCG monitoring chair for the detection of cardiac condition (Rohen, 1985). Recordings of BCGs were made over two years from normal health volunteers at Shell Headquarters in London, from students of the University exercising in the Physical Education Centre at Fenners, Cambridge and from subjects attending the mild hypertension clinic at one of the Cambridge surgeries. During the trials, for each subject, 1500 points of the BCG waveform were recorded from the seat of the BCG chair and a reference ECG signal was recorded simultaneously at a sampling rate of 100 Hz. The R-waves in the ECG signal were used to segment the BCG signal into individual beats. Altogether, 339 BCG recordings have been subjected to the analysis described in the following sections. During the trial, general practitioners of all subjects participating in the trial had been contacted and issued with a clearly marked envelope which was inserted into relevant medical records kept in the surgery. 10 cards have been returned within 24 months of the BCG recording. Of these 6 records belonged to subjects who died suddenly due to myocardial infarction. Zhao (1994) first analysed these data using ANNs and the results indicated that the chair BCG signal can find an application in predicting those subjects who will later suffer serious cardiac events before any symptoms become clinically overt..

2.4.2 Data Pre-processing

The BCG signal pre-processing method has been established in the Medical Informatics Unit, led by Dr Hanka, over years of research effort. Because of the random behaviour of biological processes and the complexity of transducer techniques employed to provide sufficient sensitivity during the acquisition of signals, various kinds of noise and interference are inevitably included in the BCG. To remove this noise, a phaseless recursive digital filter implemented by Pynsent and Hanka (1982) was adopted as a band pass filter with zero phase shift to cut off noise outside the signal band (Zhao, 1994). As discussed by Zhao (1994), this filtering procedure is based on an assumption that the signal and noise are separable in the frequency domain. In an experimental study, Zhao (1994) has shown that the phaseless recursive digital filter outperforms the backpropagation network based filter for noise removal in BCG signal and concluded that this digital filter is an effective method for getting rid of unwanted noise.

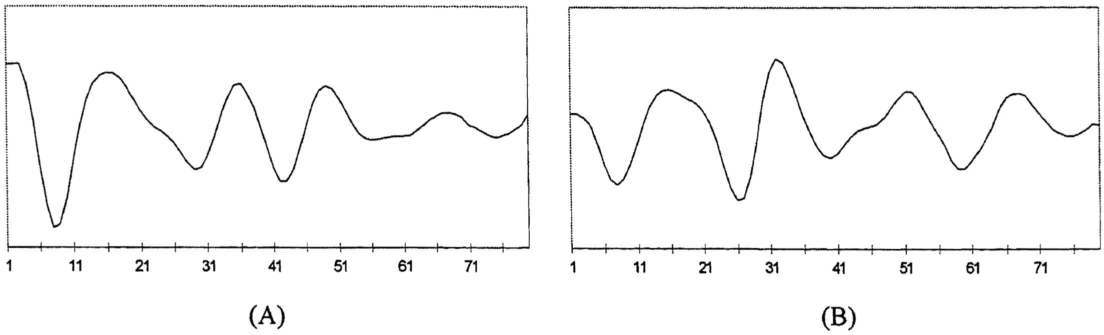


Figure 2.3: BCG mean of the normal (A) and hypertension (B).

As far as the variation of heart beats is concerned, the individual BCG beats vary from each other even within one recording. The most common method to keep the constant information contained in each beat is to appropriately average the BCG beats (Noordergraaf and Spier, 1962, Zhao, 1994). However, due to differences in the cardiac cycles of individual people, the averaged BCG beat vectors have different lengths. It is not realistic just to stretch or compress the length of every mean linearly to a standard length because the constriction of ventricles takes more or less constant time over a wide range of heart beat rates (Zhao, 1994). Rohen (1985) studied this problem extensively and developed an algorithm of non-linear adjustment of the length. She chose 80 points as the standard length of BCG. This approach is based on a dynamic time-wrapping (DTW) algorithm developed by Moor (1984) for the standardisation of speech signals. Rohen (1985) improved the DTW method by using local cross-correlation in place of DTW and producing a smooth translation table for each possible length of the BCG (between 26 and 100). The standardisation is done by first looking up the path corresponding to the length of the given mean and then adjusting the sample points accordingly. The averaged BCG means are stretched into the same length of 80 points which will be used for the classification. The standard BCG is obtained from the following equation (Zhao, 1994),

$$y_j = x_i \qquad (2.17)$$

where x_i is the averaged BCG with length of len, and y_j is the standard BCG of 80 samples($j = 0,\ldots,79$), the subscript i is $i = (\text{int})(surface(len, j))$. The $surface(len, j)$ is the pre-calculated length adjustment table given by Rohen (1985). Figure 2.3 shows the standard length BCG signal of normal and hypertension subjects. It shows that it is difficult to distinguish the features of these signals by visual inspection.

As indicated at section 1.4 of chapter 1, the same BCG data used by Zhao (1994) were used for the rest of this study. The sampled BCG signals had been processed using the above data pre-processing method. That is, the signal first passes through a phaseless recursive filter and then 10 BCG beats are averaged for noise removal. Then the averaged BCG signal is stretched or compressed to a standard length for comparison.

2.4.3 The Architecture of the Proposed Classification Scheme

As discussed in the first section, the aim of this study is to investigate the performance of the proposed neural classification system for real time application. PCA is adopted for BCG feature extraction due to its capability that it extracts information by finding the directions in input space in which the inputs exhibit most variation (Jollife, 1986) and then it projects the inputs onto the subspace to perform a dimensionality reduction. The features are then presented to a neural classifier whose function is to partition the space of features into decision regions. The proposed classification system is summarised in Figure 2.3. It has a two stage architecture. First, the PCA network is trained with training data off-line using the unsupervised learning rule described in section 2. Then, the PCA network weights are frozen and the BCG signal features are extracted from the training data. These features are then used to train the MLP classifier off-line. Next, the two resulting networks with fixed architecture and weights are incorporated into an on-line feedforward automatic classification system. As the outputs of the PCA networks have unbounded ranges, it is necessary to normalise the PCA networks outputs to values between 0 to 1.

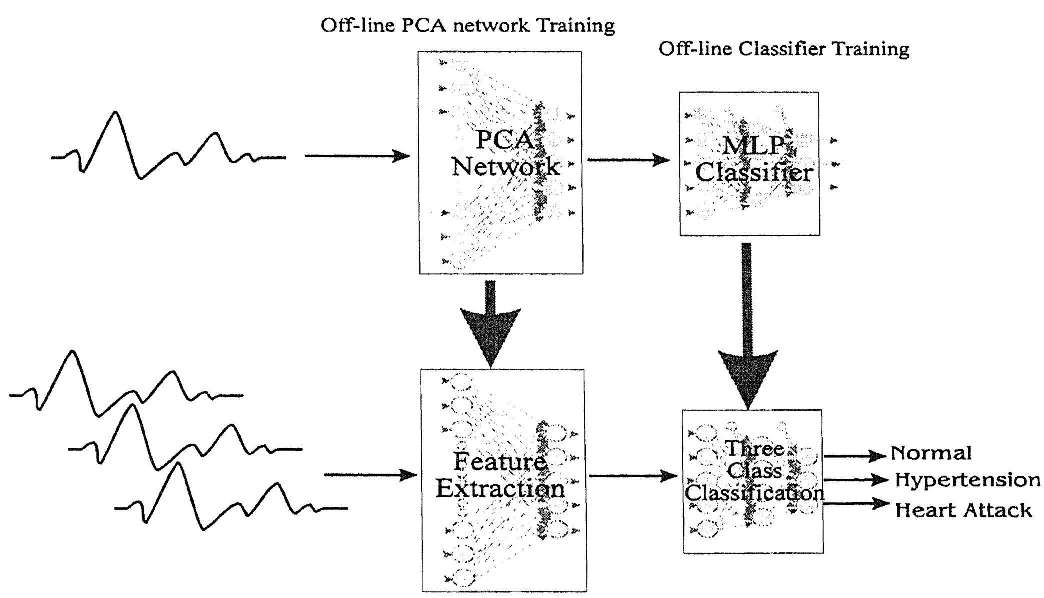


Figure 2.4 The proposed two stage network architecture for BCG automatic classification

As it has been discussed before, the amount of data to work with is relatively small. The technique for best using the available data to estimate the classification performance is usually the leave-one-out method (Devijver & Kittler, 1982). This method leaves one sample out of the whole data set and trains with the rest of the samples at each time. Then, the left out sample is used for test, and the whole process repeated again until all of the data samples have been tested. Each design and test may be regarded as independent. The disadvantage of this method is that it involves excessive computation for the whole process. On the other hand, it is of great interest in engineering applications to evaluate the classification system generalisation ability. In this study, both leave-one-out and data partitioning scheme experiments were conducted. In the leave-one-out scheme, the whole data set was used to train PCA network and then the features were used to train the neural network classifier using leave-one-out strategy. In the data partitioning scheme, the BCG data was randomly divided into two data sets: a training data set and a testing data set. The training set consisting of nearly half of the data was used to determine the necessary parameters required for both each networks. The independent set of test data was used to test the predictive accuracy of the classification system. All the simulation experiments were performed on a 486-33 MHz Personal Computer with single floating point precision and the code was compiled in Microsoft C/C++ V1.0.

2.4.3.1 Networks Training Strategies

Unsupervised PCA network training uses whole time domain BCG waveform as its input. As Zhao (1994) has shown that 3 features derived from K-Y transform are sufficient for neural classifier, the PCA networks with 5 and 10 outputs were adopted in the study. In the training phase, each vector in the training set is presented sequentially to the input of the PCA network in random order. The goal of the learning rule is to produce an output which maximally represents the variations in the input patterns. Before learning, in this study the weights were randomly initialised in a range between -0.2 and 0.2. During training, the learning rate was gradually decreased in the form of $15/(200+i)$, where i is the iteration number. For the linear PCA network, G = (1.1, 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.3). In several experiments, it has been experienced that the termination criterion was found around 1500 to 2000 iterations.

During the feedforward network training, the target values used for the output units were set to 0.0 and 1.0 respectively. The training used fixed values for the learning rate and momentum, and these were determined experimentally as 0.25 and 0.5 respectively. The initial weights were randomly set between -0.5 to 0.5. Logistic sigmoidal activation functions were used for hidden and output neurons. A bias neuron with default value of 1 was employed in the input layer and hidden layer. Three output neurons were used; one for each of the three classes, i.e. one neuron representing the normal subjects, one for the hypertension subjects and one for the patients with potential heart attack.

In MLP classifier training, the backpropagation learning rule was used and the connection weights were updated after each training example. Thus, an epoch is the number of sets of training data presented to the network between weight updates. During the learning , the mean square error is monitored against the number of presentations of the training set. An appropriate point is chosen such that no significant improvement in the mean square error is seen. The network was then trained and stopped at this chosen point. After training, the network weights were frozen and both training data set and testing data set were used to evaluate the network performance. The classification rule is based on the simple and most commonly used rule, the "winner takes all" rule which assigns the pattern to the class whose corresponding output neuron has the highest value (Rumelhart et al., 1986).

2.4.3.1.1 Training Network with Leave One Out Method

In this experiment, all of the BCG records were used as a training set for PCA network. In order to obtain 'zero mean data' as required for principal component analysis, the mean vector was estimated from the whole training set and subtracted from each data vector. Two PCA networks (linear and non-linear) were implemented The linear PCA network used in this experiment was trained with the modified Oja's rule that has homogeneity and locality. A non-linear PCA network with the learning rule of the equation (2.6) was also explored. The non-linear output is defined by $z_i = \tanh(\beta y_i)$. PCA networks with 5 and 10 outputs were evaluated. For the non-linear PCA networks, the gain of the transfer function β was tried with values 1,2 and 3.

Once the PCA network training process was completed, the corresponding converged weights of the outputs were used to reduce 80 inputs pattern to just 5 or 10 components. These components were normalised into a range between 0 and 1. To ease the comparison, the number of hidden nodes implemented in the classifier were defined according to the experiment results described in section 2.4.3.1.2. For 5 principal component outputs, the number of hidden nodes was fixed to 4, and for 10 outputs, the number of hidden nodes was fixed to 8. The normalised PCA components were used to train the neural classifier with the leave-one-out method. This method has been applied to the training of neural network classifiers in small sample size situation (Martin et al., 1994; Zhao, 1994). It has been shown that this method can provide nearly unbiased estimates (Devijver and Kittler, 1982; Ney et al., 1995). For a training set of N vectors, the training scheme can be summarised as follows:

1. Drop one BCG record out of the data set
2. Design the classifier based on the remaining N-1 training vectors
3. Test with the vector that was dropped out
4. Repeat the step 1, until the N vectors were tested

However, this method is very expensive in computation because it needs N times training process. For a 10:8:3 network, 144 hours were required for one training run with a 486DX-33 PC.

2.4.3.1.2 Training Network with Partitioned Data set

Although the leave-one-out method would be best in minimising the bias of the estimate error for it makes use all available data samples, it is not practical for use in neural network training since that would require training the network thousands of times, once for each dropped feature sample. In applying the classification system to real world problem, the question of generalisation ability of the system is of great interest for solving real world problem. A popular method to understand the generalisation ability of a system is data partitioning scheme that a data set is divided into two subsets; one is called the training set and the others is called testing set. The training set is used to train the system until the different of described outputs and actual output is minimum. Elements from testing set is used for measuring the generalisation ability of the system. In order to obtain computable estimates of the generalisation performance of the combined system, data partitioning scheme was also conducted. This is essential for a system which is to be trained off-line and implemented on-line, as shown in Figure 2.4.

Table 2.1 shows the partitioned training set and testing set. The training data samples were randomly selected from the BCG data base. First, the PCA network was trained using the procedure described in section 2. The mean of the training data was estimated from each individual training data vector to obtain zero mean for the training input vectors. After training, the PCA network weights were fixed to 8-bit resolution for BCG feature extraction. The extracted features were then normalised to values between 0 and 1 which were used to train a three-layer feedforward network to produce the identity of the class. The neural classifier architecture and the parameters were the same as with the leave-one-out neural classifier.

Table 2.1: Training and Testing Data Sets

Subjects	Training Data Set	Testing Data Set
Normal	30	28
Hypertension	111	164
Risk of Heart Attack	3	3

To decide the number of hidden nodes required in the MLP, an experiment was conducted by varying the number of hidden nodes using linear PCA components. Four runs of different initial weight values were carried out for each network architecture. Each run of the training process was fixed to 5000 iterations. One of the best performance of all the runs was chosen for training data and testing data evaluation. Figure 2.5 shows the results of the overall misclassification rate as a function of the number of hidden nodes for 5 inputs MLP and for 10 inputs MLP respectively. It is noted that with the 5 inputs MLP classifier, four hidden nodes gave a low misclassification rate in both training and testing data sets compared with 3 hidden nodes and 5 hidden nodes. A low misclassification rate was achieved for the 10 input MLP, when the number of hidden nodes was set to 8. Thus, the number of hidden nodes were fixed to 4 and 8 for 5 input MLP and 10 input MLP respectively.

The number of iterations used in training the MLP was estimated from a two pass procedure; where in the first pass the mean square error is monitored against the number of iterations. An appropriate point is chosen such that no significant improvement in the mean square error is seen, and all the training was repeated up to this chosen point. Once the training was finished, the network weights were fixed and the networks were tested with the testing data set. Since the backpropagation algorithm is sensitive to different starting points, the classifier training was carried out with 4 runs starting from different random initialisation for the weights of the network. The best performance network was chosen for the final test. The same procedure was used for both the 5 and 10 component outputs.

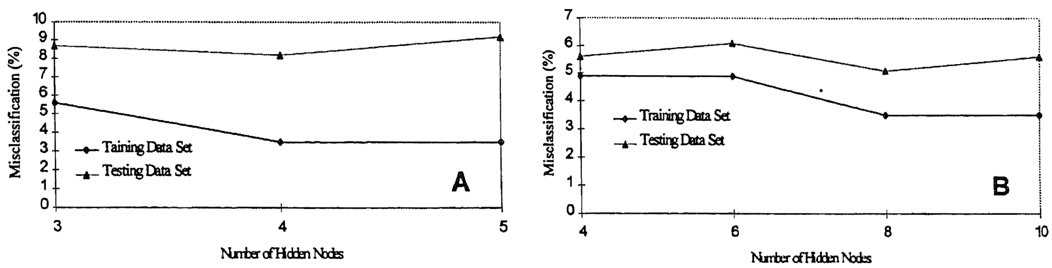


Figure 2.5: Misclassification rate at a function of the number of hidden nodes for (A) 5 inputs MLP classifier and (B) for 10 inputs MLP classifier.

2.4.4 The Effects of Noise on the Network Performance

As mentioned in section 2.4.4 that BCG data used in this study has been pre-filtered and averaged to improve the signal-to-noise ratio, the evaluation of the classification system performance is based on a relatively 'noise free' condition. In real world application, it is unavoidable that the BCG signal could be severely corrupted. Therefore, the evaluation of the robustness of the proposed classification system in extreme cases could provide an insight for future robust system design. It has been reported that the influence of respiration and of abrupt shifts in baseline caused by body movement during the measurement are the most serious sources of noise for the chair BCG signal (Brown et al., 1952). The signal pre-processing method developed in the Cambridge University might not be effective in removing the severe body movement noise because the 10 signal averages might not be enough. In this study, abrupt movements and the respiration were selected as representative noise sources. Movement of the patient while recording the BCG was simulated by adding a DC bias to a given segment of the BCG. The maximum level was 25% of the maximum amplitude of the uncorrupted BCG. This is equivalent to a noise level of 250% in one BCG signal before averaging 10 of them. Figure 2.6 - (B) shows the plot of the BCG corrupted by the abrupt shifts in the baseline. Respiration interference was simulated by adding a low frequency sinusoid signal to the BCG means. The frequency was 0.33 Hz (Friesen et al., 1990) and the maximum amplitude was 25% of the maximum amplitude of the uncorrupted BCG. This could be considered a noise level around 180% in one BCG signal before averaging 10 of them. A plot of the BCG signal corrupted by baseline drift due to respiration is given in Figure 2.6-(A).

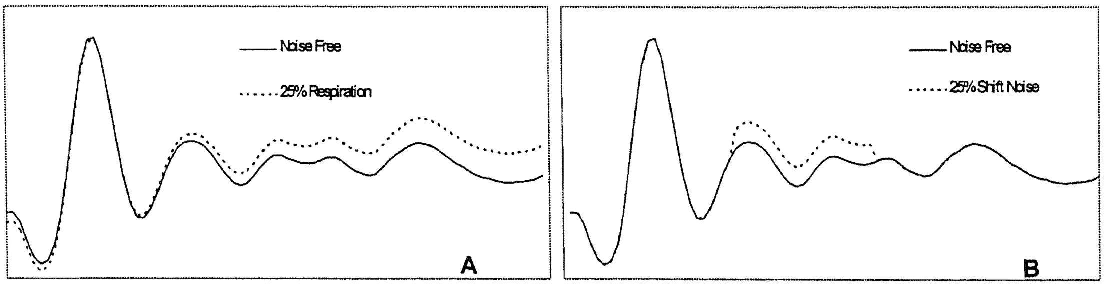


Figure 2.6: BCG corrupted with baseline drift due to respiration (A) and abrupt shift (B).

Many studies of artificial neural networks have noted that adding noise (e.g. Gaussian noise) to the training data could improve the network generalisation and fault tolerance (Holmstrom and Koistimen, 1992; Jim et al., 1996). However, researches have also shown that a small learning rate and a large number of training samples are required to average over the noise to achieve the good generalisation results (Reed et al., 1995; Peterson et. al., 1995). The study in chapter 3 shows that the small number of sample patterns of the heart attack class cause a bias estimation problem. Thus, adding noise to BCG signals is only used for the trained network test in this chapter.

2.5 Experiment Results

Several simulation experiments were conducted to evaluate the network performance. The classification results are used to build a confusion matrix of a table whose each row represents the actual outputs of the network.

2.5.1 Leave One Out Training Scheme

The results of the leave-one-out experiment are summarised in the following confusion matrix tables.

Table 2.2: Classification Results Using 5 Principal Components (Linear PCA Network)

Class	Normal	Hypertension	Heart Attack
Normal	47	11	0
Hypertension	5	270	0
Heart Attack	1	5	0
Overall Performance	93.5%		

Table 2.3: Classification Results Using 5 Principal Components (non-linear PCA Network, $\beta=1$)

Class	Normal	Hypertension	Heart Attack
Normal	48	13	0
Hypertension	4	271	0
Heart Attack	2	4	0
Overall Performance	94.1%		

Table 2.4: Classification Results Using 5 Principal Components (non-linear PCA Network, β=2)

Class	Normal	Hypertension	Heart Attack
Normal	48	10	0
Hypertension	4	271	0
Heart Attack	1	3	2
Overall Performance	94.7%		

Table 2.5: Classification Results Using 5 Principal Components (non-linear PCA Network, β=3)

Class	Normal	Hypertension	Heart Attack
Normal	40	19	0
Hypertension	12	263	0
Heart Attack	1	3	2
Overall Performance	90%		

The experimental results for the non-linear PCA networks show that the best performance was achieved for β=2. It was considered that using 5 principal components might not provide sufficient information. Thus, the experiment was carried out with the output components increased to 10. For the non-linear PCA network, only the training with β=2 was implemented because it provided good performance. Table 2.6 shows the performance of linear PCA with 10 output components, and Table 2.7 shows the non-linear PCA network with β=2.

Table 2.6: Classification Results Using 10 Principal Components (Linear PCA Network)

Class	Normal	Hypertension	Heart Attack
Normal	54	4	0
Hypertension	6	269	0
Heart Attack	1	5	0
Overall Performance	95.3%		

Table 2.7: Classification Results Using 10 Principal Components (non-linear PCA Network, β=2)

Class	Normal	Hypertension	Heart Attack
Normal	51	7	1
Hypertension	5	265	5
Heart Attack	1	1	4
Overall Performance	93.8%		

2.5.2 Partition Training Scheme

From the leave one out training results, it is noted that the non-linear PCA network with β=2 gives the best performance in classifying the heart attack subjects. It is also shown that the 10 principal components have more discriminate information for the neural classifier. In the data partition scheme, we only consider the linear PCA network and non-linear PCA network with β=2 which gives the best result. It is considered that the performance error can be used to compare with the results of the leave-one-out method.

Table 2.8: Classification Results Using 5 Principal Components (Linear PCA Network)

Class	Training Data			Testing Data		
	N	H	HA	N	H	HA
Normal (N)	28	2	0	21	7	0
Hypertension (H)	0	111	0	6	158	0
Heart Attack (HA)	0	3	0	0	3	0
Overall Performance		95.1%			91.8%	

Table 2.9: Classification Results Using 5 Principal Components (Non-Linear PCA Network, $\beta=2$)

Class	Training Data			Testing Data		
	N	H	HA	N	H	HA
Normal (N)	28	2	0	20	8	0
Hypertension (H)	0	111	0	4	160	0
Heart Attack (HA)	0	3	0	0	3	0
Overall Performance		95.1%			92.3%	

Table 2.10: Classification Results Using 10 Principal Components (Linear PCA Network)

Class	Training Data			Testing Data		
	N	H	HA	N	H	HA
Normal (N)	28	2	0	23	5	0
Hypertension (H)	0	111	0	2	162	0
Heart Attack (HA)	0	3	0	0	3	0
Overall Performance		96.53%			94.87%	

Table 2.11: Classification Results Using 10 Principal Components (Non-Linear PCA Network, $\beta=2$)

Class	Training Data			Testing Data		
	N	H	HA	N	H	HA
Normal (N)	30	0	0	22	6	0
Hypertension (H)	0	111	0	3	160	1
Heart Attack (HA)	0	1	2	1	2	0
Overall Performance		99.31%			93.33%	

2.5.3 Noise Data Testing

In this study, previously trained combination of 10 outputs linear PCA network and MLP, and combination of 10 outputs non-linear PCA network with MLP were evaluated. Simulated respiration and abrupt noises describe in section 2.4.4 were added to the 80 points BCG data. All the parameters in the networks were kept the same as before. The testing data set results are shown in Table 2.12 and 2.13 respectively.

Table 2.12: Classification Results Using 10 Principal Components (Linear PCA Network)

Class	Respiration noise test			Abrupt noise test		
	N	H	HA	N	H	HA
Normal (N)	22	6	0	21	7	0
Hypertension (H)	4	160	0	7	157	0
Heart Attack (HA)	0	3	0	0	3	0
Overall Performance		93.33%			91.28%	

Table 2.13: Classification Results Using 10 Principal Components (Non-Linear PCA Network, β=2)

Class	Respiration noise test			Abrupt noise test		
	N	H	HA	N	H	HA
Normal (N)	21	7	0	23	5	0
Hypertension (H)	3	161	0	2	162	0
Heart Attack (HA)	0	3	0	0	3	0
Overall Performance		93.33%			94.87%	

2.6 Discussion

2.6.1 Performance Analysis

With the available data set in hand, combined networks with 5 and 10 features have been tested with a leave-one-out evaluation scheme. From results in Table 2.2, 5 features derived from the linear PCA have poor performance on the heart attack class. In the variation of the transfer function gain of the non-linear PCA network experiments, it is illustrated that the best performance was achieved with $\beta = 2$ at which 2 of the heart attack subjects were correctly classified. Comparing Table 2.4 with Table 2.5, shows that further increasing the β value did not improve the classification performance of heart attack subjects, but decreased the performance of normal and hypertension classification.

When the number of features was increased to 10, the results in Table 2.6 and Table 2.7 show that the classification accuracy of the linear PCA improved on the normal and hypertension subjects, but still could not classify the heart attack subjects. This might be caused by unbalanced training patterns in the training set. During the network training, the small number of heart attack class patterns could not affect the weight change efficiently. However, a significant improvement was achieved with the non-linear PCA as shown in Table 2.7. 4 heart attack subjects were successfully classified. This result demonstrates that the change to non-linear PCA projection can provide good discriminate information in this particular problem. These results also suggested that 5 features could not provide enough information for the neural classifier and that 10 features have the ability to maintain the required information. In a previous study using the K-Y transform for BCG signal feature extraction, the classification results achieved with a 3 inputs, 3 hidden units and 3 outputs network trained with the leave-one-out scheme is shown in Table 2.14 (Zhao, 1994). This result shows that features generated by the K-Y transformation can yield superior performance by considering the class variance and between-class variance. However, our results show that for the classification accuracy of the normal and hypertension subjects, the features derived by the linear or non-linear PCA network have competitive performance compared with K-Y features. The poor performance of the heart attack subject for the non-linear PCA features might be caused by unbalanced class patterns in the training set. Thus, it is expected that the classification performance on this class can be improved when the available samples are increased.

Table 2.14: Classification Results Using K-Y Features with a MLP [3:3:3] (Zhao, 1994)

Class	Normal	Hypertension	Heart Attack
Normal	55	3	0
Hypertension	2	273	0
Heart Attack	0	1	5
Overall Performance		98.23%	

2.6.2 Generalisation Capability

Comparing the results of 5 feature classification in Table 2.8 and Table 2.9, and the results of 10 feature classification in Table 2.10 and Table 2.11, it is suggested that the feature dimensionality has a significant effect on the generalisation capability of features generated by a linear PCA network, but much less on that of the non-linear PCA network. For the heart attack subjects, even the features generated using non-linear PCA could not be classified. This suggests that the small distribution of class patterns in the training data set could not provide enough information to code network weights. Overall, the experimental results suggest that features generated by linear PCA and non-linear PCA networks provide similar generalisation capability on normal and hypertension subjects. The poor generalisation for the heart attack subject might be caused by the small number of class patterns presented in the training samples. Thus, it is expected that the generalisation ability of the combined neural classification system would be improved if the class patterns in the training set were balanced and a large number of training data samples were available.

The study of leave-one-out and data partitioning results shows that the standard statistical methods for BCG feature extraction are not the optimal methods for this problem when only a limited amount of data is available. Theoretically, the statistical method needs a large training set to provide reliable statistical estimates, and the test set must be drawn from the same probability distribution as the training set (Raudys & Jain, 1991). Unless a special modification is applied (Hoffbeck and Landgrebe, 1996), it is not ideal to use the standard statistical methods for feature extraction in this study. Although the K-Y transform method illustrates good results in feature extraction by using the whole data as training set, the generalisation capability of this method is not evaluated. With such a small data set, it is not expected that good generalisation results could be achieved. It is suggested that standard statistic methods should be avoided in BCG feature extraction study unless the data set is big enough.

2.6.3 Effect of Noise in Testing Data

The robustness of the proposed classification systems have been evaluated by presenting noise in the testing data set. It is noted that respiration noise and abrupt noise have a small effect on the combined classification system performance. The combined linear PCA and neural classifier system has better tolerance to respiration noise than to abrupt movement noise. The combined non-linear PCA and neural classifier system shows robustness against both noise types. This result suggests that the non-linear PCA network is more robust than the linear PCA. Thus, the combined non-linear PCA network and neural classifier system is a good candidate for real time application, especially in a noisy environment. Although the performance of the network to the noise data can be significantly improved by using noisy data to train the network, the limited training data set could decrease the performance, as will be discussed in the network pruning section in chapter 3.

2.7 Conclusions

In this investigation, motivated by the study conducted by Zhao (1994), an alternative classification system based on combined neural network approaches is proposed to classify normal, hypertension and heart attack subjects, aiming at on-line BCG classification in real time. The neural network implementation of PCA is adopted for feature extraction. Both leave-one-out and data partitioning methods were implemented to evaluate the proposed classification system. In the leave-one-out training experiment, it is shown that for the combined linear PCA network and neural classifier system, the misclassification rate for the heart attack class is 100%, but for the combined non-linear PCA and neural classifier system, the classification accuracy for this class was improved to 66.7%. It is suggested that about 10 features will be required to maintain classification accuracy for the proposed combined neural network system for this particular problem. In the partitioned data experiment, the generalisation ability of the combined neural network systems were evaluated. The results suggest that the classifier is sensitive to the dimensionality of features generated by linear PCA network but less dependent to the dimensionality changing of features which are generated by non-linear PCA network. The results indicate that the number of data samples and balance of the class patterns in the samples have a significant influence on the generalisation capability of the combined classification systems. Finally, the robustness of the proposed systems was tested with noise added to the testing data set. The results show that the combined non-linear PCA network and neural classifier system has better noise tolerance ability than the combined linear PCA system.

The overall results of this study show that the statistical method has potential capability for BCG feature extraction in real world application if the training samples were adequate. However, it is not an optimal method for this study with such limited amount of data samples, since the statistical method needs a large training data set to provide reliable statistical estimates, for example, to accurately estimate a sample covariance matrix (Hoffbeck and Landgrebe, 1996). Although the K-Y transform methods show that a better result can be achieved by making use of the class assignment information in the whole data set, it has not been tested for generalisation performance. Furthermore, the factor of collecting the heart attack subject data is a difficult and time consuming task. It is expected that shortage of heart attack subject data will always persist in the studies. Therefore, it is concluded that one should avoid directly using standard statistical methods for the feature extraction studies with a limited data set, as the limited data set could cause errors in statistical estimation and degrade the classifier performance. A further investigation of non-statistical methods for BCG signal dimensionality reduction is required to meet real time application requirements. In the next chapter, a method using a multiresolution wavelet transform method for BCG signal dimensionality reduction is explored.

Chapter 3

BCG Classification Using Wavelet Transform and Neural Networks

3.1 Introduction

In chapter 2, the combined neural network approach for BCG signal classification has been investigated. The neural extension of PCA was adopted for BCG feature extraction. The results show that feature extraction based on the standard statistical method could bias the generalisation result because the numbers of available data samples is too small to provide enough information for statistical estimates. Because of the difficulty and expense involved in collecting heart attack subject data, it is desirable to develop a BCG feature dimensionality reduction method using the available data set without involving the statistical feature extraction procedure. Thus, a multiresolution analysis method for BCG feature dimensionality reduction is explored in this chapter.

Among the methods in the literature for signal processing, a new transform method, the wavelet transform, has become a very important and effective tool in signal and image processing. Unlike Fourier transform that represents a function as superposition of sines and cosines, the wavelet transform contains more complicated basis functions called wavelets. This allows translating a time domain function into a representation that is localised not only in frequency but in time as well. In the last few years, the wavelet transform has proved useful in a variety of medical signal and image processing applications, including ECG signal compression (Thakor et al., 1993; Chen et. al., 1993), ECG diagnosis analysis (Crowe et. al., 1992) and EEG signal analysis (Petrosian et al., 1995). The combined wavelet transformation and neural network approach has been shown to be a robust solution for complex, real world applications (Shukla et al., 1993; Moor and Merany, 1995; Szu et al., 1996).

An attractive feature of wavelet transformation is its suitability for extracting information from the original signal at various scales (Daubechies, 1988; Mallat, 1989). The basic idea behind the multiresolution analysis is to split up the given signal at different resolutions and to analyse them separately. This idea is justified by the fact that the information at different resolutions generally characterises different physical structures of the signal (Mallat, 1989). Therefore, these independent frequency channels can be treated according to their different physical properties. The multiresolution representation can be viewed as passing a signal through a sequence of filters with successively narrower bandwidths. This approximation process results in a set of successively lower resolution representations of the original signal. The information lost at each stage of the smoothing process is retained in a detail signal (Mallat, 1989). If the correct scale of wavelet multiresolution analysis is chosen, it can provide the information of interest with reduced dimensionality. Therefore, it is possible to classify the data in a compressed format to reduce the data dimension and to save the computation time. Motivated by this idea, the multiresolution wavelet analysis is used here for BCG signal dimensionality reduction. A small subset of the wavelet coefficients are used as the input to the neural classifier. To the best of the author's knowledge, this is the first time that wavelet transforms have been applied to BCG signal analysis.

The remainder of this chapter is organised as follows: In section 2, following a brief background review of discrete wavelet transforms, the method of BCG signal dimensionality reduction using a fast discrete wavelet transform algorithm is implemented. In section 3, a combined fast wavelet transform algorithm and neural network classification system is proposed and evaluated. The simulation results are presented in section 4. Based on the simulation results, a comparison with the combined statistical method described in chapter 2 is discussed in section 5. Finally, the conclusion is given in section 6.

3.2 Multiresolution Analysis

3.2.1 Discrete Wavelet Transform

The essential idea of a wavelet analysis is to choose a generation function, the wavelet, and to give an associated transformation of a time-scale representation of functions called the wavelet representation. In a wavelet representation, continuous wavelet transformation (CWT) corresponds to both time and scale parameters being continuous, and discrete wavelet transformation (DWT) corresponds to both time and scale parameters being discrete (Rioul and Vetterli, 1991). DWT has attractive features in practical applications that use digital computers to transform sampled signals. The detailed mathematical aspects of DWT are discussed by Daubechies (1988) and Mallat (1989). This section briefly describes the multiresolution representation of a function using orthonomal wavelets. That is, the families of wavelet functions are orthogonal and normalised. Consequently, the signal is decomposed into pairs of signals from which the original could be reconstructed.

The specific features of the multiresolution wavelet are that wavelet functions and scale functions at successive scales are orthonormal. The functions can be defined by a pyramidal processing that decomposes the signal of interest $f(x)$ into its coarse component $C_{j,n}$ and detail component $D_{j,n}$ at various scales 2^j, where j is a scale parameter and n is a position parameter. The coarse component $C_{j,n}$ represents the approximation of the original signal $f(x)$ with a resolution of one point per every 2^j points of the original signal. The detail component $D_{j,n}$ represents the details of the original signal at the corresponding levels of resolution (Figure 3.1).

Coarse features of a function $f(x)$ are localised with the aid of a scaling function $\phi(x)$. This function can be selected to decompose the signal over a dyadic scale, $\phi_{j,n}(x) = 2^{-j}\phi(2^{-j}x - n)$. The scaling function assists in the localisation of the coarse features:

$$C_j(n) = < f(u), \phi_{j,n}(u) > \qquad (3.1)$$

where the notation $< f(u), \phi_{j,n}(u) >$ refers to an inner product $\int_{-\infty}^{+\infty} f(u)\phi_{j,n}(u)du$ which is the convolution of the input signal with the scaling function at scale 2^j and position n.

The detail components of the function $f(x)$ can analogously be retrieved by employing a wavelet function $\psi(x)$ called mother wavelet which is orthogonal to $\phi(x)$, again selected to decompose the signal of interest over a dyadic scale. The detail components can be obtained by generating a series $\psi_{j,n}(x) = 2^{-j}\psi(2^{-j}x - n)$. The wavelet function assists with the extraction of the detail features:

$$D_j(n) = < f(u), \psi_{j,n}(u) > \qquad (3.2)$$

Mallat (1989) has shown that the tree or pyramid algorithm can be adapted to the DWT using the wavelet coefficients as the filter coefficients to apply to a raw data vector. One filter works as a low pass filter to obtain the resolution change, and one filter works as high pass filter that to bring out the data's detail information. The low-pass filter coefficients are associated with the scaling function $\phi(x)$. The high-pass filter is associated with wavelet function $\psi(x)$. The computation of the coarse and the detail components from discrete samples can be obtained by the following convolutions:

$$C_j(n) = \sum_{k=-\infty}^{\infty} \tilde{h}(2n-k)C_{j-1}(k) \qquad (3.3)$$

$$D_j(n) = \sum_{k=-\infty}^{\infty} \tilde{g}(2n-k)C_{j-1}(k) \qquad (3.4)$$

where k is the sample index of the inputs to the filters and n is the sample index of the outputs from the filters. Mallat (1989) shows that for the filter bank algorithm (Figure 3.1), the functions and filters coefficients are related by :

$$h(n) = < \phi_{1,0}(u), \phi(u-n) > \qquad \tilde{h}(n) = h(-n) \quad (3.5)$$

$$g(n) = < \psi_{1,0}(u), \phi(u-n) > \qquad \tilde{g}(n) = g(-n) \quad (3.6)$$

More details on wavelet multiresolution analysis can be found in Mallat (1989) and Daubechies (1988). The specific example of determination of the filter coefficients proposed by Mallat (1989) is reproduce in Appendix A.

3.2.2 Multiresolution Representation

In the BCG signal classification, most of the information needed for classification is in the BCG shape and related parameters, such as the wave amplitudes of H, I, and J, slopes of base line to H, H-J, and I-J, and times of H-J, I-J as well as H-I and H-J in percent of heart period (Harrison et al., 1967). These kinds of features are often dependent on the degree of resolution. At a coarse resolution, information about the large features of the complex waves is represented; while at a fine resolution, information about more localised features of individual waves is formed. Upon discarding selected detail, one can obtain the required information for the classifier without much signal distortion. It is true that not every resolution is equally important to provide the classification information in the classification phase. Therefore, it is possible to choose a proper resolution and discard the selected component without much signal distortion. Using the wavelet transform, one can project the raw BCG data into a small dimension space and decompose the original signal into a series of sub-signals at different scales. The coarse components which still have the significant shape information can be used to present to the neural network classifier.

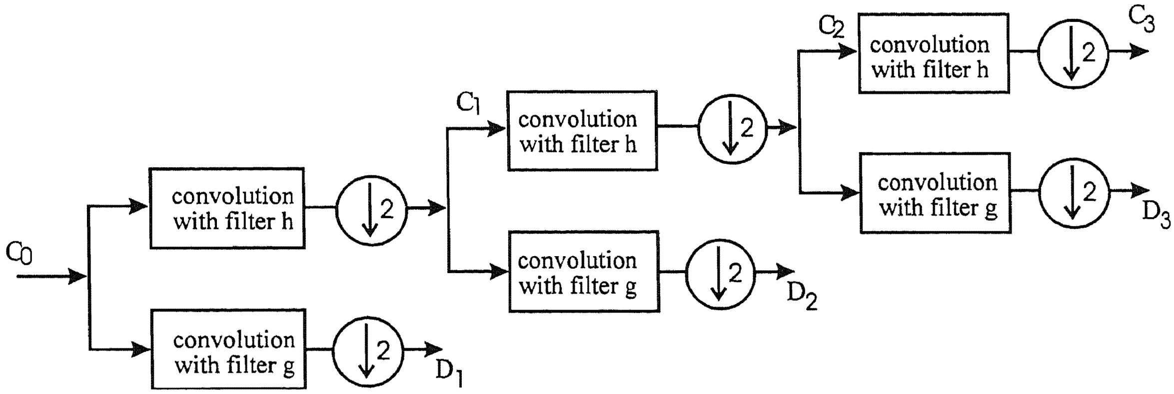


Figure 3.1 Wavelet decomposition algorithm (Mallat, 1989).

One of the most useful features of wavelets is that one can choose the defining coefficients for a given wavelet system to be adapted for a given problem. In Mallat's original paper (1989), he developed a specific family of wavelets that is good for representing polynomial behaviour. As there is no prior information to suggest that one wavelet function would be better suited to BCG analysis than others, the filter coefficients derived by Mallat (1989) are adopted for this study. The algorithm for calculating the coefficients constructed by Mallat (1989) is given in Appendix A. In this study, the signal reconstruction procedure is not required. Thus, only low pass filter processing is used to approximate the original BCG signal. The fast discrete wavelet transform algorithm described by Cody (1992) is implemented in C code for BCG signal decomposition.

As the original data frame has 80 sample points, at each successive resolution level, the pyramidal algorithm compresses the data by a factor of 2. Figure 3.1 shows the three level multiresolution wavelet decomposition algorithm. At the first process level, the N dimension signal $f(x)$ is taken as C_0 and is decomposed into two bands: coarse component $C_{j,n}$ and detail component $D_{j,n}$ which both have N/2 samples. At the next level, only the low-pass output signal is further split into two bands. Signals at various resolutions give information about different features. Figure 3.2 shows three levels of coarse coefficients derived by using the fast wavelet transform algorithm. In time scale, they are 40, 20, and 10 samples respectively.

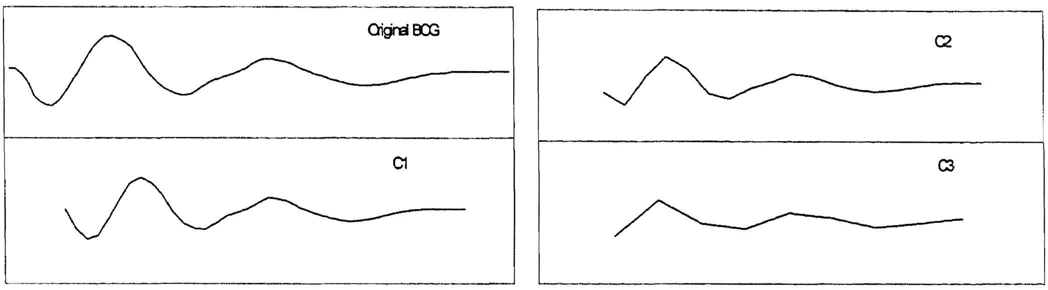


Figure 3.2 Coarse components of the BCG waveform at each level

The multiresolution representation process results in a set of successively lower resolution representations of the original BCG signal. Notice that the 40 and 20 coarse components have kept most of the original waveform features, and the 10 coarse components have totally lost the H wave which might have classification information for this study. It a previous study, it has been shown that the useful frequency range of these BCG signals is within 0-20 Hz (Yu et al., 1996). During the forward wavelet transformation, the pyramid algorithm divides the spectrum into octave bands. As the BCG signal sampled at 100 Hz, the low pass bands covered by wavelet coefficients ***C1***, ***C2*** and ***C3*** represent frequency cut off at 25 Hz, 12.5 Hz and 6.5 Hz respectively. It is obvious that some of the spectrum components have been attenuated for 20 coarse coefficients and more spectrum components are lost for 10 coarse coefficients. However, due to the feature of wavelet transform that allows translating a time domain function into a representation that is localised not only in frequency but in time as well, the localised waveform features, such as wave complexes, could still be preserved for classifications even though some high frequency information has been lost. To verify this point, three experiments are constructed to train the neural networks using 40, 20 and 10 coarse components respectively. Each neural network is trained with the leave-one-out method. The preliminary classification results showed that the performance with 20 coarse components has same performance to that of 40 coarse components on heart attack subjects which has a misclassification rate of 33.3%. However, 10 coarse components give a misclassification rate of 66.6% on heart attack subjects. In this study, 20 coarse components are used for the neural classifier inputs.

3.3 Description of the Combined Wavelet Transform and Neural Network Classification System

The proposed classification system is illustrated in the Figure 3.3. The on-line classification system is based on two processing phases. A two level wavelet transform is implemented. The original BCG signal is passed to the first low-pass filter which subsamples the inputs by a factor of two. Then, the low-pass filter output is used as the data input for another low pass filter, identical to the first one, generating another set of approximation coefficients at the second level of scale. At the output of the second level, the 20 approximation coefficients are then normalised into the range between 1 and 0 as the neural classifier inputs. A MLP classifier is applied to classify these wavelet coefficients into associated classes. The network architecture and the weights were determined off-line as stated in chapter 2.

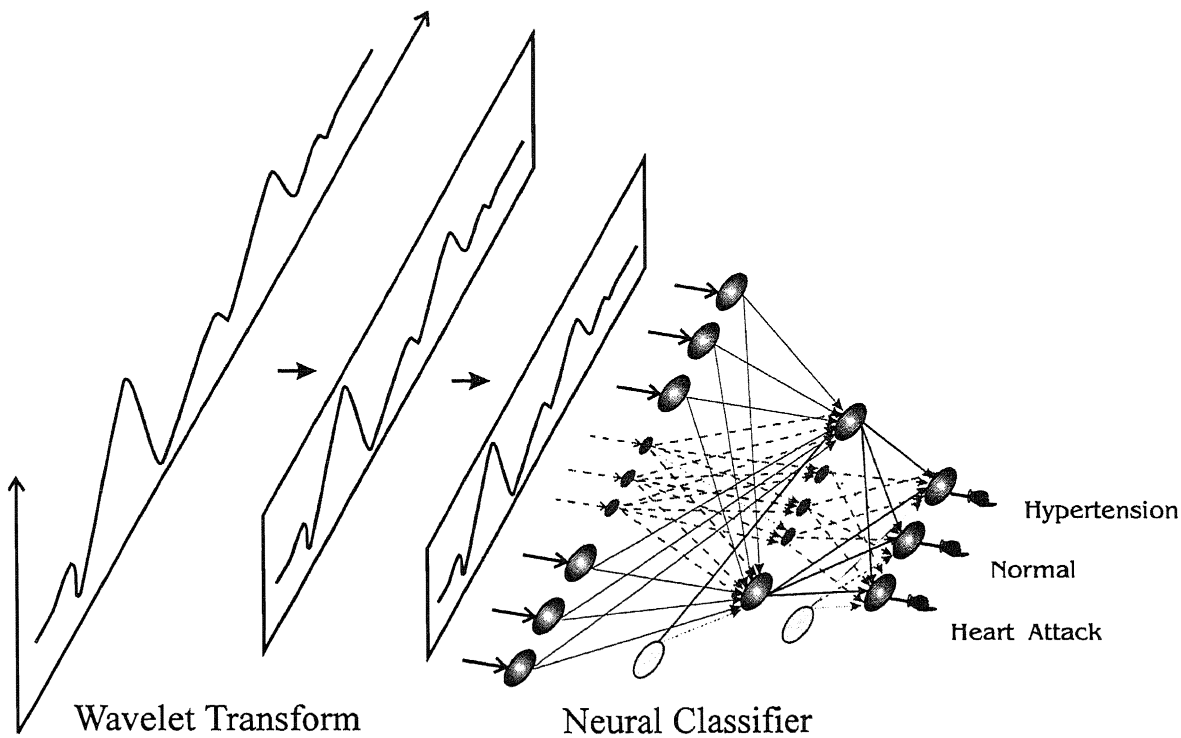


Figure 3.3: The combined BCG classification system

In order to make a fair comparison with previous results, both the leave-one-out and data partitioning schemes described in the chapter 2 were evaluated. The previous partitioning of the data set was used for the training data set and testing data set. The partitioning data scheme was used to decide the number of hidden units required. The training procedure and parameters were the same as in chapter

2. The performances were evaluated on both the training data set and the testing data set. The number of hidden nodes used to construct the classifier were decided empirically. Tables 3.2 to 3.6 show the performance results with variations of the number of hidden units from 8 to 3. Based on the results, the number of hidden nodes was fixed to 4 for the rest of the study.

3.4 Experimental Studies in BCG Classification

3.4.1 Results of the Network Trained with Leave One Out Scheme

Based on the experimental results in section 3.4.2, the number of hidden neurons were fixed to 4. The performance of training set and testing set with leave-one-out scheme are shown in Table 3.1.

Table 3.1: Classification Results Using Leave-One-Out Training Scheme

Class	Normal	Hypertension	Heart Attack
Normal	54	4	0
Hypertension	2	271	2
Heart Attack	0	2	4
Overall Performance	97.1%		

3.4.2 Results of the Data Partition Scheme

The same partitioning of the data set was used in all runs of this investigation. The wavelet coefficients were truncated to 8-bit resolution and normalised into the range between 0 and 1. It was observed that 4000 iterations were adequate for network convergence. The generalisation capability of the neural classifier was evaluated using the same strategy. The results for variations in the number of hidden nodes are shown in Table 3.2, Table 3.3, Table 3.4, Table 3.5 and Table 3.6 respectively. The results show that the neural classifier with 4 hidden nodes gives the best generalisation performance on the testing data set. Therefore, the network with 4 hidden nodes is adopted in the data partitioning scheme.

Table 3.2: Classification Results Using 8 Hidden Nodes

	Training Data			Testing Data		
Class	N	H	HA	N	H	HA
Normal (N)	30	0	0	24	3	1
Hypertension (H)	0	111	0	5	158	1
Heart Attack (HA)	0	1	2	0	1	2
Overall Performance	99.3%			94.36%		

Table 3.3: Classification Results Using 6 Hidden Nodes

Class	Training Data			Testing Data		
	N	H	HA	N	H	HA
Normal (N)	30	0	0	24	3	1
Hypertension (H)	0	111	0	3	158	3
Heart Attack (HA)	0	1	2	0	1	2
Overall Performance		99.3%			94.36%	

Table 3.4: Classification Results Using 5 Hidden Nodes

Class	Training Data			Testing Data		
	N	H	HA	N	H	HA
Normal (N)	30	0	0	24	3	1
Hypertension (H)	0	111	0	3	159	2
Heart Attack (HA)	0	1	2	0	1	2
Overall Performance		99.3%			94.87%	

Table 3.5: Classification Results Using 4 Hidden Nodes

Class	Training Data			Testing Data		
	N	H	HA	N	H	HA
Normal (N)	30	0	0	25	2	1
Hypertension (H)	0	111	0	2	159	3
Heart Attack (HA)	0	1	2	0	1	2
Overall Performance		99.3%			95.38%	

Table 3.6: Classification Results Using 3 Hidden Nodes

Class	Training Data			Testing Data		
	N	H	HA	N	H	HA
Normal (N)	30	0	0	24	3	1
Hypertension (H)	0	110	1	5	156	3
Heart Attack (HA)	0	1	2	0	1	2
Overall Performance		99.3%			93.3%	

3.4.3 Pruning Network with Weight Decay Technique

The classification system presented in this chapter is based on the combination of DWT and MLP. The inputs to the neural classifier are a low resolution representation of the BCG waveform. It would be useful to examine whether some features could be irrelevant or less relevant to this classification task. Pruning is one of the practical methods for network architecture optimisation (Hertz et al., 1991). It has a clear advantage of improving the network generalisation capability by removing unnecessary weights or even nodes. Furthermore, removing the unnecessary weights or nodes can also reduce the computation complexity for real time implementation. One common technique is the weight decay method. This technique augments the usual least squares cost function E with a function which is the sum of the square of all the weights in the network (Krogh & Hertz, 1992; Plaut & Hinton, 1987):

$$E = \frac{1}{2}\sum_{p}\sum_{i}(o_{ip} - d_{ip})^2 + \lambda\frac{1}{2}\sum_{i} w_{ij}^2 \qquad (3.7)$$

This cost function leads to a simple "weight decay" term in the weight update:

$$w(t+1) = w(t+1) - \lambda w(t) \qquad (3.8)$$

This is easy to implement, and limits the development of any weights with "large" absolute value and the weights that are unimportant for the task will decay to zero at a rate determined by λ. Thus, it is claimed that it can prevent the network from over-fitting the data and minimises the sensitivity of the output to noise in the input (Nowlan & Hinton, 1992). It leads the network to generalise well. In this experiment λ was set to 0.0001. After the training, only 13 out of 99 total weights had a magnitude close to zero. The results for the application of weight decay to the network with $\lambda = 0.0001$ are shown in Table 3.7. The generalisation for the hypertension class was slightly improved on the testing data set, the normal subject classification was not improved and the heart attack cases were totally misclassified. Compared with results for the network trained without weight decay, the overall classification accuracy had dropped. Apparently, the penalty term in the weight decay method limited the weight growth in early stages of training. If the population of the pattern in the training sample distribution has a significant difference, the weight update value of the small distributed class patterns in the training data samples will become even weaker due to the penalty term of weight decay. This can actually destroy the ability of the training process to find complex localised structure in the small class patterns. Therefore, a large samples are needed in order to be confident that the correct correspondence was not pruned. Since the performance for heart attack subjects was poor after only a small number of weights were removed, no further pruning experiments were made to further reduce the weights.

Table 3.7: Classification Results with Weight Decay Parameter $\lambda = 0.0001$

	Training Data			Testing Data		
Class	N	H	HA	N	H	HA
Normal (N)	20	0	0	24	4	0
Hypertension (H)	0	111	0	4	160	0
Heart Attack (HA)	0	3	0	0	3	0
Overall Performance		97.22%			94.4%	

3.4.4 Robustness to Noise

To test the robustness of the combined wavelet transform and neural classifier system, noise was added to the testing data set as discussed in chapter 2. The results are summarised in Table 3.8. Many studies have noted that adding small amounts of noise to the training data often results in improvement generalisation and fault tolerance (Sietsma & Dow, 1988; Weigend et al., 1991; Holmstrom and Koistinen, 1992; Chakrabarti et al., 1995). This improvement is important for real world applications.

Thus, an experiment was conducted by adding 25% abrupt shift noise and 25% respiration noise to the training data separately. Table 3.9 shows the performance of the testing data result.

Table 3.8: Results of Adding Noise to the Testing Data Set

Class	Added respiration noise			Added abrupt shift noise		
	N	H	HA	N	H	HA
Normal (N)	23	4	1	24	3	1
Hypertension (H)	8	150	6	16	138	10
Heart Attack (HA)	0	1	2	0	0	3
Overall Performance		89.7%			79.5%	

Table 3.9: Classification Results of the Network Trained with a Noisy Data Set

Class	Added respiration noise			Added abrupt shift noise		
	N	H	HA	N	H	HA
Normal (N)	21	6	1	23	4	1
Hypertension (H)	5	156	3	7	152	5
Heart Attack (HA)	0	1	2	0	1	2
Overall Performance		91.8%			90.8%	

3.5 Discussion

3.5.1 Performance Analysis

3.5.1.1 Comparison of the Proposed System with Previous Systems

The objective of this study is to develop a reliable, simple and fast system which can perform well with the limited data set that is to hand. The property of the wavelet decomposition of signals is to identify features that are localised in a waveform (such as H, I, J and K peaks etc. in Figure 1.2) and in time scale. It is considered that this method is a kind of feature extraction method that takes the wave complex into account. Thus, the feature extraction is achieved by projecting the waveform onto a low resolution representation to keep necessary information. In this study, the original dimensionality was reduced by 75%.

Table 3.2 shows that performance estimated using the leave-one-out method is competitive with that of the combined K-Y transform and MLP system proposed by Zhao (1994). By using the data partitioning scheme, Table 3.6 shows that the best results were achieved on the testing data set by using a 20-4-3 architecture, especially for the heart attack class. Overall, both leave-one-out and data partitioning results demonstrated that the combined wavelet transformation and MLP classification system outperforms the combined PCA and MLP networks system. The generalisation capability for the small class of heart attack patterns was improved. In view these results and the fact that the total number of examples is not very large, the combined wavelet transform and MLP classification system can be said to perform well compared to the results discussed in section 2.6.1.

Research suggests that selecting an appropriate training data set can improve the network generalisation capability (Plutowski and White, 1993; Hasegawa et al., 1996). If the training set is not ideal, it will not make the network generalise well. As the training data set and testing data set was randomly selected, it is difficult to be sure that the training data set covers all. An experiment was conducted by swapping the training data set with the testing data set. With 4 hidden nodes, the experimental results were as shown in Table 3.10. The results in the experiment indicated that the generalisation performance of the network was little changed. It is concluded that the partitions of the training data set and testing data set generally have similar distributions.

Table 3.10: Classification Results of Swapped Data Sets

	Training Data			Testing Data		
Class	N	H	HA	N	H	HA
Normal (N)	28	0	0	27	3	0
Hypertension (H)	0	164	0	0	107	4
Heart Attack (HA)	0	1	2	0	1	2
Overall Performance		99.49%			94.44%	

3.5.1.2 Comparison of the Wavelet Transform and Signal Re-sampling Scheme

In order to reduce the neural network size, Zhao (1994) has tried the effect of resampling the BCG data at 50 Hz to reduce the number of network inputs to 40. A network with 40 inputs, 5 hidden nodes and 3 outputs is trained with the leave-one-out method. The results are shown in Table 3.11. Comparing this classification result to the Table 3.1, it is shown that the classification result using wavelet coefficients is superior to the signal resampling method. The BCG signals have a frequency range 0-20 Hz. According to the sampling theorem, it is not possible to downsample the BCG signal to 20 points. However, the experimental results indicate that the wavelet transform is capable to reduce the BCG dimensionality to 20 points for classification. This provides an attractive alternative for BCG signal analysis.

Table 3.11: Classification Results Using Leave-One-Out Training Scheme with 40 Inputs (Zhao, 1994)

Class	Normal	Hypertension	Heart Attack
Normal	49	9	0
Hypertension	6	266	3
Heart Attack	0	4	2
Overall Performance		93.51%	

3.5.1.3 Visualisation of Data Distributions

In order to analyse the original data and reduced data distribution visually, the linear PCA mapping algorithm described in section 2.2.2.1 is utilised, which projects all data samples onto a two dimensional plane which contains components in the first and second eigenvectors. This will give the directions with large variances in the distribution of both original BCG data and wavelet transformation output. (Yu et. al., 1996). Figure 3.4 shows distribution maps of the original BCG data and wavelet transform output in **A** and **B**. It shows that the distribution of all patterns before and after wavelet transformation have similar directional distribution structures. The pattern after wavelet transformation is a more compact in the vertical direction and extended in the horizontal direction compared to the map of the original signals. There is a great similarity between the two maps. This suggests that much significant information is preserved after wavelet transformation.

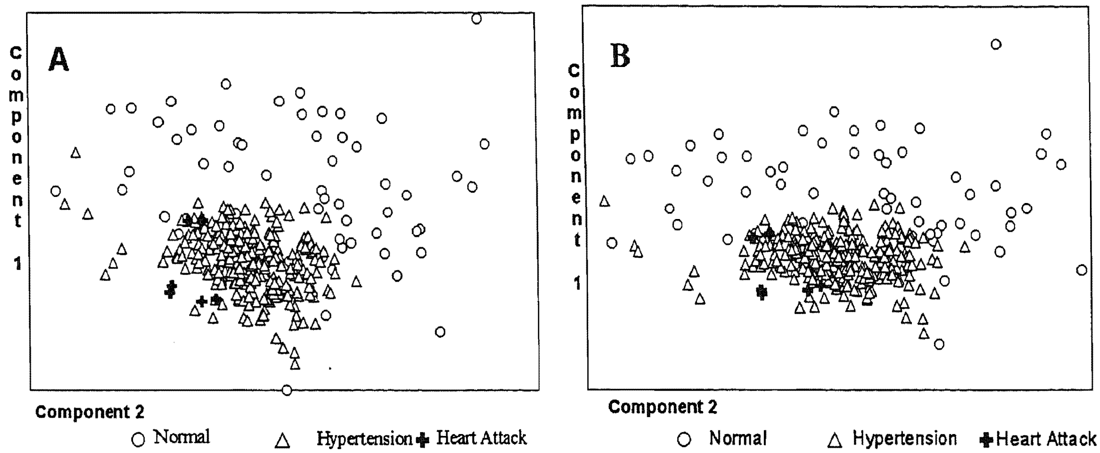


Figure 3.4: Two principal components map of (A) original BCG signal and (B) 20 coarse components.

3.5.1.4 Pruning Network

A common weight decay method was implemented in an attempt to improve the network generalisation by removing unnecessary weights. The experimental results suggested that it might be difficult to improve the network generalisation performance with small training samples. Apparently, the penalty term in the weight decay method limits the weight growth in the early stages of training and can actually destroy the ability of the training process of finding complex localised structure in the data if the class patterns in the training samples is small. This result confirmed the points made by Hasagawa et al. (1996) that if the training set is not ideal, it is difficult to improve the network generalisation by using network generalisation improvement methods.

3.5.1.5 Effect of Noise

The effect of noise on the proposed classification system was evaluated. As shown in Table 3.9, the results show that the performance decreases dramatically when the classifier is tested with noisy data, especially for the abrupt noise caused by baseline shift. This is probably caused by the fact that the classifier relies on the wave shape and wave complexes information. When the waveform is corrupted with noise, the classifier cannot properly classify the corrupted waveform according to the original information learned. Compared with the results of the combined PCA and MLP network system, the combined wavelet transform and neural classifier system has worse performance on the noise test. The combined PCA and MLP network system has a large number of weights which can provide a good fault tolerance capability. To improve the noise tolerance of the combined wavelet transformation and MLP system, the method of adding noisy data in the training data set to train the network was investigated. Table 3.10 shows the testing results when the networks were trained with respiration noise and abrupt shift noise added to the training data set. The results of this investigation show that the overall performance of the trained network on the noisy test data set was significantly improved. However, the normal and heart attack classes had no significant improvement. This was probably caused by the small number of training patterns. Reed et al. (1995) have pointed out that a large number of training samples and a small learning rate are required to average the noise. The experimental results suggest that adding noise to the training data set is an effective method for improving the network fault tolerance. This method should be considered in future to improve the robustness of the proposed classification system when a moderate number of training samples are available.

3.5.1.6 Computation Efficiency

For real time applications, the reduction of computation complexity is of greatest concern for the engineering design. Table 3.12 shows a comparison of the forward pass computation complexity of the two combined classification systems. It shows that the computation operation required for the combined wavelet transformation is similar to that of the combined PCA network with 10 outputs. However, the memory storage required for the combined wavelet transform and neural classifier system is much smaller. Moreover, unlike the PCA networks, no training procedure is required to generate the features for classifier inputs. This provides a great advantage for real time implementation. The comparison results of performance and computation efficiency suggest that the combined wavelet transformation and MLP system is suitable for real time implementation.

Table 3.12: Comparison of the Computation Complexity for BCG Feature Extraction

	Combined PCA Method			Combined Wavelet Method		
	Multiplication	Addition	Storage	Multiplication	Addition	Storage
Feature Extraction	800	808	808	720	660	12
Classifier	115	115	115	99	99	99
Total	915	923	923	819	759	111

3.5.2 Implementing an Automatic BCG Classification System

In view of the studies of the previous chapter and this chapter, an automatic system for chair BCG classification in real time is proposed as illustrated in Figure 3.5. The system mainly consists of three function blocks; signal pre-processing, wavelet transform and neural classifier. Two channels of real time signals from sensors are connected to the on-line classification system. The R wave peak detected by the R-wave trigger circuit is used to identify the BCG cycle (Richez, 1992). The incoming BCG signal first passes though a high pass analogue filter to reduce the respiration noise level. Then the signal is digitised at 100 samples/second with 8-bit resolution. The records of 10 beats are averaged by accumulation to improve the signal to noise ratio. The averaged BCG record is stretched or compressed to a standard length of 80 points by making use of a standardisation lookup table. The stretch or compress procedure is described in section 2.4.2 of chapter 2.

A two level fast wavelet transform is applied to the 80 points of the signal pattern to reduce the dimension to 20 points. Although constructing a compact supported mother wavelet involves complex mathematics, the calculation of the wavelet transform is relatively simple. The wavelet transform can be implemented as finite impulse response (FIR) filters as shown in Figure 3.1. In this specific application, only low-pass filters are needed. The 20 points signals are then classified into three classes; normal, hypertension or risk of heart attack by a neural classifier. The architecture and weights of this on-line neural classifier are determined using the off-line training method that is described in chapter 2. The implementation of these three function blocks could be either pure software solution or the combination of software and hardware approaches. As discussed in chapter 1, one of the primary goals of this study is to investigate implementing neural network on hardware so that the instrument can quickly process real time information. Thus, a hardware implementation of the neural classifier for this specific application is adopted for the proposed system implementation.

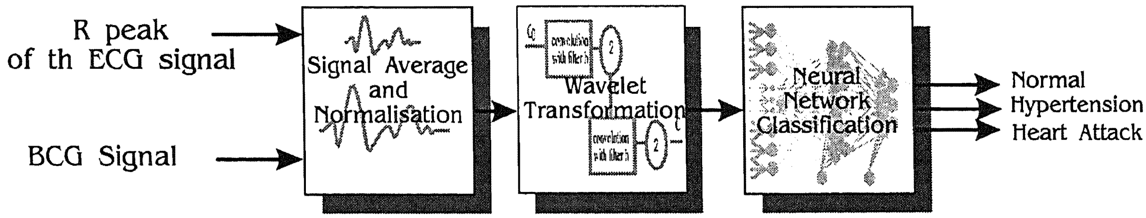


Figure 3.5 The block diagram of the proposed automatic BCG classification system

3.6 Conclusions

This chapter proposed a new method for BCG classification. It consists of a two stage process: the multiresolution wavelet transform for BCG feature reduction and a neural classifier to partition the space of features into decision regions. Since the characteristics of BCG signals are mainly features of shape and time interval between peaks, these kinds of features often depend on the degree of resolution. Multiresolution analysis offers the practical solution to this problem. Mallat's spline wavelet (Mallat, 1989) is used in BCG signal decomposition. The simulation results using the leave-one-out training scheme show that the proposed classification system has better performance than a combined neural networks approach, and has comparable performance with the combined K-Y transform and neural classifier method proposed by Zhao (1994). The data partitioning scheme test showed that the combined wavelet transformation and MLP classification system has better generalisation than the combined neural networks system (Yu, 1996). Two common methods for improving network generalisation and robustness are evaluated; one is to use a weight decay technique, and the other is to present noise to the training data set. The simulation results also suggested that the improvement is constrained by the limited number of data samples. Comparing to the BCG signal resampling scheme used by Zhao (1994), it indicates that the wavelet transform is an effective approach in BCG signal dimensionality reduction. Furthermore, it has been demonstrated that the wavelet transform has a compact computational complexity. From the classification performance and real time implementation point of view, the multiresolution wavelet is promising for an on-line dimensionality reduction in BCG analysis. Finally, an automatic classification system is proposed for real time BCG signal classification. The proposed classification system incorporates signal pre-processing, wavelet transform and neural classifier modules. The implementation of the proposed system can be either pure software or the combination of software and hardware approaches. In this study, a hardware implementation of the neural classifier for this specific application is adopted to address one of the goal of the project.

Chapter 4

Developing a Hybrid Learning Algorithm for Weight Precision Selection

4.1 Introduction

In chapter 3, a combined wavelet transform and MLP technique was proposed that can achieve the goal of automatic BCG classification for real world application. In implementing neural networks to solve real world problems, a hardware implementation of the neural network is usually required to meet the speed and space constraints needs (for example, bioelectric potentials, such as ECG, BCG, EEG, and biomedical image processing). However, one of the problems in the design of neural hardware is the resolution of the synaptic weights. It is not possible to implement high precision in analogue circuits, and high precision requires a big multiplier and a large memory to store weights which occupy large areas on digital chips. It has been demonstrated that a high performance can be achieved with limited precision in weights (Lont and Guggenbuhl, 1992; Barranco et al., 1993). Research also shows that the precision required for the weights varies and is dependent on the problem difficulty and on the network size (Yoo and Pimmel, 1994). Boser et al. (1992) developed a neural circuit with 5-bit weights for optical character recognition. Lisa et al. (1993) demonstrated a 2-bit weights network to solve the vehicle licence recognition problem. More recently, Leong and Jabri (1995) have implemented a neural classifier in a multilayer perceptron chip (Kakadu) with analogue technology for cardiac arrhythmia classification. The synaptic weights implemented in the chip have only 6-bit resolution. It is a common question to be considered before the neural hardware design what minimum number of bits is necessary for successful operation of the neural classifier. If the optimal trade-off is achieved, the design and power cost of the instrumentation can be significantly reduced.

By analysis of the backpropagation learning rule described in chapter 2, two phases of operations can be distinguished; forward pass and backward pass. In the forward pass, the networks take input patterns from the training sets and compute the network outputs; during the backward pass, the outputs are compared with desirable outputs to generate the error signal which is used to adjust the connection weights. Once the network is properly trained, the neural networks perform the forward pass phase to retrieve corresponding outputs. Usually, a high precision is required in the backpropagation phase to help the network to converge to the desired point. For the forward pass classification phase, it has been shown that the neural networks require less weight precision (Boser et al., 1992; Piche, 1995).

According to the training phase implementation, the hardware implementation of neural networks can be divided into two categories, on-chip training and off-chip training methods.

In the on chip training method, both phases are performed by the hardware and the weights update is operated on-line. There has been considerable work on implementing the backpropagation algorithm in dedicated hardware to speed up the computation. However, the precision required during the training is usually high (Holt and Hwang, 1993). Hollis et al. (1990) show that 12-bit resolution is required to learn a non-linear simple problem and a typical suggested resolution is 16 bits. This results in extensive hardware area and complexity. It is possible to relax the required weight precision with other learning methods such as weight perturbation (Jabri and Flower, 1992). Sakaue et al. (1993) illustrated a weighted error function learning algorithm for digital number recognition with 8 bit resolution. However, the cost in these methods for achieving learning with low precision weights is an increase in the complexity of the training algorithms. Therefore, these algorithms usually need complex circuits to implement them for on-line learning which also means that a large silicon area and power consumption are required for digital implementation.

In most of the real word applications using neural networks, the neural networks are trained on serial computers off-line, with possible later implementation in hardware once a satisfactory set of weights is found. In the off-line training method, both phases are simulated on a digital computer. All the network parameters are determined during the simulation. During the training, the weights are quantized and the performance is evaluated until the network reaches a satisfactory performance. The final weights are fixed and then downloaded into hardware processors specialised for forward pass operation. This method provides the capability to devise algorithms for very low precision weight representation. Thus, a dedicated hardware implementation of such a neural network can have improved processing speed and reduced circuitry complexity and power consumption. One disadvantage of this method, however, is that it might require a longer training cycle than the on-line training method.

The approach of off-line training with high precision, on-line implementation with reduced precision is useful in practical applications, as in most real world cases the neural classifiers need only one training with sufficient data samples and can then be used for a long time. Thus, the development of a general methodology for the implementation of such systems in a specific technology(e.g. VLSI) can reduce the circuit complexity, operate much faster, and save cost. The work presented in the following chapters belongs to this category.

As the weight precision required depends on the specific task, it is necessary to investigate the variation in diagnosis accuracy with the weight precision in this BCG classification task. Moreover, a special learning scheme is required so as to obtain compact weight precision. This chapter is devoted to the limited precision weight learning problem. A training algorithm for weight precision selection is introduced for real world applications. This chapter is organised as follows: following a brief review of the related works in limited weight precision learning on section 2, a new hybrid learning algorithm is proposed in section 3 and a comparison with the standard backpropagation (BP) method is conducted using two benchmark problems. In section 4, the proposed learning algorithm is used for precision reduction in the BCG neural classifier. Finally, a discussion and conclusions are presented in section 5 and section 6 respectively.

4.2 Related Works

Reduction in weight precision is an active area in artificial neural network research. Nakayama et al. (1990) proposed an algorithm that trains networks with high precision using the standard backpropagation rule and then gradually reduces the number of precision bits during retraining cycles. It is found that this method can reduce weight precision significantly compared with conventional backpropagation method with discrete weight resolution. It has been demonstrated that this method can reduce the weight precision to 2-bit for a digital number recognition problem. Lisa et. al. (1993) reported a learning algorithm that introduces a penalty term to the backpropagation cost function. The network is first trained with high resolution weights with the penalty term set to zero. When the network converges, the penalty term is increased gradually during the retraining phase. Thus, at the final stage, a network with low resolution weights and simple activation function can be obtained. However, the above methods have one common problem of convergence in that the network can become stuck in a local minimum during learning, due to limited weight precision and rounding operations.

In order to overcome this problem, Dundar and Rose (1995) proposed a "backpropagation with quantization" algorithm. In this algorithm, the network is trained with high resolution. When the training is completed, the weights are quantized and the network is retrained. If the network becomes paralysed, the updates for all the weights are registered. The n largest updates are chosen and the least significant bits of the weights associated with these updates are flipped. As a result, the network can avoid local minima created by the process of quantization. Alibeik et al. (1995) developed a learning algorithm by combining quantized weight learning and weight perturbation. The network is initially trained with high resolution. When the training is completed, the weights are quantized and the network is trained with a quantized gradient descent rule (QGDR) to avoid local minima. The weights (W_i) are updated in the opposite direction of $\partial E / \partial W_i$ for an amount equal to one quantization step. This change is accepted if it decreases the network error function E, otherwise it is rejected. This process is applied to all of the weights in the network until the error function becomes less than a predefined value or no further change in the weights are made in one epoch. It was shown that the QGDR algorithm can reduce the weights precision to 2-bit for the Arabic digit recognition problem.

In addition to the modified backpropagation algorithms for discrete weight learning, a new type of weight perturbation method and associated modification algorithms for low precision learning have been introduced (Jabri & Flower, 1992; Leong & Jabri, 1995; Maeda et. al., 1995). The characteristic of these algorithms is that they estimate the gradient of the error with respect to a certain weight, by perturbing the weight and dividing the resulting change in the error by the amount of the weight perturbation. In order to obtain the required weight precision, several learning algorithms have been proposed. It has been shown that these algorithms can train the network with reduced precision. However, Dundar and Rose (1995) have pointed out that computational complexity depends more strongly on network size with these methods than with standard backpropagation. Therefore, these methods are not efficient for training large networks. The backpropagation rule is arguably the most popular network training algorithm in off chip learning to achieve the lowest resolution weights because of its simplicity and its popularity for multilayer feedforward neural network learning (Dundar and Rose, 1995).

4.3 A Hybrid Learning Algorithm for Fixed Point Precision Network Training

4.3.1 Finite Precision Computations

In fixed precision learning, the weight values have a given $q+1$ bits fixed point form, while expressed as one sign bit plus $q-n$ bits integer and n bits to the right of the point with proper range which depends on the problems associated. In general, the rounding or truncating error due to the limited precision in the learning phase can be described in the following two cases:

1. If values require more than q bits precision, but are less than $2^{(q-n)}$ in magnitude, the rounding/truncation operation will chop the lowest order bits off.
2. If values are less than 2^{-n} in magnitude, they will be treated as zero.

Because of the truncating operation in fixed precision learning, low precision weights could cause the problem that most of the weights might be truncated to zero especially when the learning rate is small. In this case, the backpropagation error signal described in chapter 2 (2.10) and (2.12) could be very small, and no weights updating can be applied even though the difference between the actual outputs and desirable outputs are still high (Reyneri & Fillippi, 1991, Dundar and Rose, 1995). The performance improvement of the classification drops to very low levels or even approaches zero. One practical method to deal with this problem is to use a large learning rate during the training (Dundar et al., 1995). However, if the learning rate is too large, the derivative $f'(x) = f(x)[1-f(x)]$ could vanish due to the rounding operation and the learning procedure again becomes stuck (Salomon & Hemmen, 1996). Once a neural network is stuck, escaping from this state using gradient based learning is extremely difficult, because the amount of weight update on the neuron is near zero. Thus, the network is paralysed. In the following section, an addition is made to the standard backpropagation algorithm to improve the network convergence for low precision learning.

In this study, inputs and outputs are 8-bit values to the right of the point with range [0.0, 1.0]. The bias value θ is an 8 bit value with range [0.0, 1.0]. The weight values are expressed as $q+1$ bits fixed point form. During the simulation, truncation is used after each weight update calculation.

4.3.2 The Combined Training Algorithm

To tackle the network paralysis problem, a modified random search approach is added to the standard backpropagation algorithm to prevent the conventional BP rule from stopping on a paralysis problem. The random optimisation method has the property that it ensures convergence to a global minimum of the neural network error function (Baba, 1989). However, this approach has the limitation that it sometimes takes a long time for convergence. On the other hand, standard backpropagation has a good capability for local optimisation. Thus, a hybrid algorithm is proposed to make use of both the merits of the backpropagation method and the random optimisation method of Solis and Wets (1981) to find global minimum in a small number of iterations. In this proposed hybrid algorithm, the standard back-propagation algorithm is used to decrease the error function with comparatively small steps and the cost function changes are monitored after a number of iterations. If the variation of the cost function is bigger than a defined value, the backpropagation rule is used to continue to update the weights. Otherwise, it means that the weight update has stopped and the network is paralysed. In this case, a random weight perturbation method is used to update the weight value to help the network to escape paralysis.

In this study, the backpropagation algorithm described in chapter 2 was used without a momentum term. The network weights were updated after every example. As elsewhere in this study, the mean squared error (MSE) function E was used:

$$E = \frac{1}{pm}\sum_{p}\sum_{m}(d_{pm} - o_{pm})^2 \qquad (4.1)$$

where m is the number of output nodes, and p represents the number of input patterns.

The random optimisation method proposed by Solis and West (1981) was modified to accommodate fixed precision calculation. Instead of using a Gaussian function to generate the random numbers, random values were chosen from within the quantization step interval. The modified random optimisation method for the network weights update is defined as following:

1. Generate random values which are equally distributed in the interval ($-\Delta$, $+\Delta$), where Δ is a quantization step.
2. If $E(w_i + \Delta) < E(w_i)$, then $w_{i+1} = w_i + \Delta$.
3. If $E(w_i + \Delta) > E(w_i)$ and $E(w_i - \Delta) < E(w_i)$, then $w_{i+1} = w_i - \Delta$.
4. If $E(w_i + \Delta) > E(w_i)$ and $E(w_i - \Delta) > E(w_i)$, then $w_{i+1} = w_i$.

The criterion of gradient threshold (GT) and error threshold (ET) proposed by Leung et al. (1994), is adopted in this study to check whether a paralysis is met. If the network is paralysed, the random optimisation method is applied to help the network convergence. Otherwise, the backpropagation learning rule is used for weights update. The modified learning algorithm can be summarised as following:

Step 1: Initialise the random weight values.

Step 2: Perform a forward pass and calculate the system error of the n-th iteration E_n.

Step 3: If E_n < defined stop criterion, then stop; otherwise, perform a backward pass: update the weights.

step 4: Check the iteration steps. If the iteration steps have not reached the defined number of iterations, go to step 2; otherwise go to step 5.

Step 5: Compute the error gradient ΔE by:

$$\Delta E = (E_n - E_{(n-t+1)}) / t$$

where t is the number of iterations for estimation of ΔE.

Step 6: If ($|\Delta E|$ < gradient_threshold and (E_n > error_threshold), it means that no error change has occurred and the network has been paralysed. In this case, the backpropagation rule cannot adjust the network weight any more. Thus, the modified random optimisation method is applied to break such minima and to help the network converge. After random weight optimisation for one epoch, go to step 2; otherwise directly go to step 2.

In this algorithm, the chosen number of iterations t is important. If a large value is chosen, the effect of the MSE changes could be ignored and the learning might be trapped in one situation for a long time. If too small a window size is used, the backpropagation algorithm cannot update the weights properly. There will be more oscillation during the network training. In this experiment, t was chosen as 10 iterations. The same procedures are repeated several times, until the MSE is less than the default value or no further improvement is achieved.

The value of GT usually depends on the weight precision defined during the training. If the weight precision is high, a small value should be chosen otherwise no error improvement is recorded in a specific window and the random optimisation method cannot be applied. The network is trained with the standard backpropagation rule only. However, if the precision becomes low, one can expect a large MSE oscillation because of the truncating error. A small GT in low precision weight training could cause the network to use the random optimisation method frequently and eliminate the backpropagation fine tune capability. Therefore, a relatively high GT should be used. Another important value is the choice of the ET. If the ET is too small for the very low precision weight training, it could affect the network convergence as the low resolution weight usually generates a slightly high error signal. If the ET is too large, the random weight perturbation will become useless and the learning rule becomes standard backpropagation. Figure 4.1 shows the training of a network on the XOR problem as described below with 10 bit weight resolution. It is shown that a large ET gives the same total error function as the backpropagation method. When the ET value is set up between 10^{-2} and 10^{-3}, the network converges well.

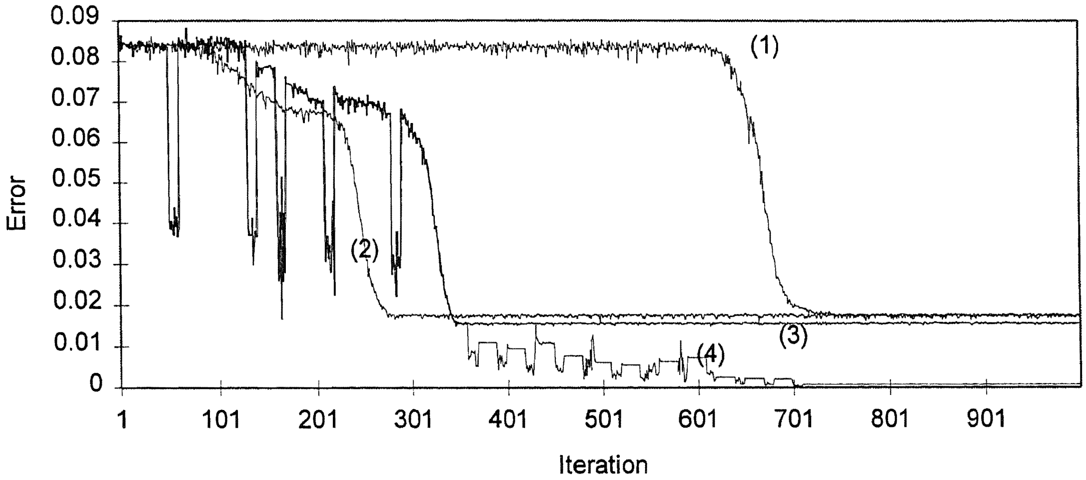


Figure 4.1: Learning curves of the 10-bit weights network on XOR problem.

(1) Back-propagation (BP) algorithm

(2) Hybrid algorithm with ET = 0.5

(3) Hybrid algorithm with ET = 0.01

(4) Hybrid algorithm with ET = 0.001

4.3.3 Benchmark Results

Two benchmark problems were used to evaluate the proposed algorithm. One is the XOR problem and the other is the classification of a two-class problem with overlapping regions. The goals in choosing the benchmark problems are that the samples should be adequate and simple enough to allow an analysis and comparison study, while on the other hand, they should be representative of real world classification problems.

4.3.3.1 XOR Problem

The XOR problem (or two-bit parity problem) is a hard classification task. It is used often as a standard benchmark to test new learning algorithms because it is a simple yet highly non-linear function which may be viewed as a special case of a more general problem (Minsky and Papert, 1969; Rumelhart et al., 1986B). Figure 4.2 illustrates the classic XOR problem. For two binary inputs x_1 and x_2, the system outputs a 1 if it receives a [0,1] or a [1,0] and responds 0 otherwise. It clearly shows that this pattern could not be linearly separated.

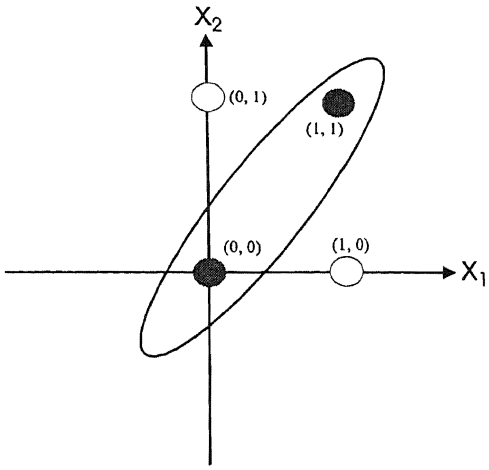


Figure 4.2: XOR problem.

Both conventional back-propagation and the proposed algorithm were used to train a neural network with 2 input nodes, 2 hidden nodes and one output node respectively. The initial weights were randomly generated in the interval of [-0.5, 0.5]. The discrete weight precision was varied from 12-bit to 6-bit. For each weight precision, 20 runs of the learning algorithm were carried out with a maximum iteration number of 1000. As suggested by Neubauer (1995), a successful training sequence was considered to have been achieved if the error criterion E was below 0.02.

All the number of bits given for the weights in this study include 1 bit for the sign bit. The corresponding convergence rate γ was then calculated as the number of successful training sequences divided by the number of all training sequences. Typical learning curves are shown in Figure 4.1, Figure 4.3 and Figure 4.4 respectively for 10-bit, 8-bit and 7-bit weight precision, using the same initial weight values.

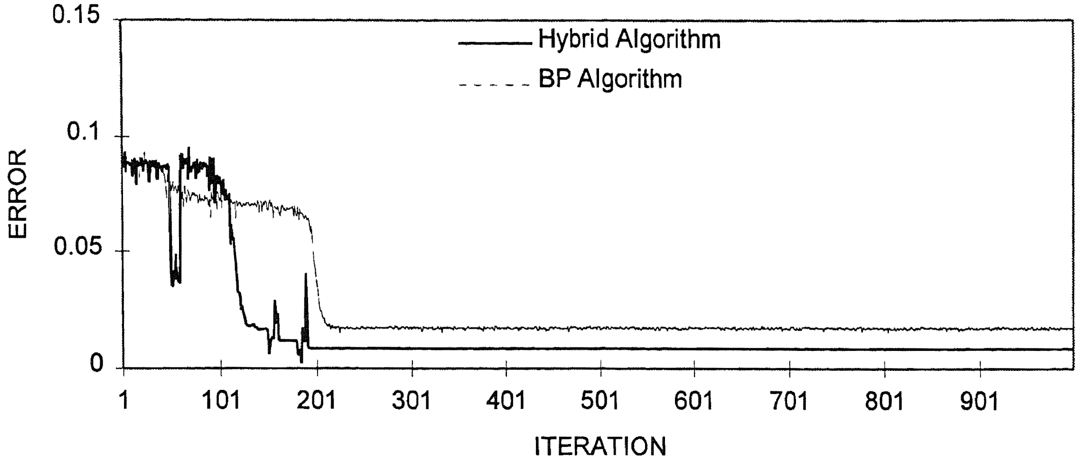


Figure 4.3: Learning curves of limited weight precision of 8-bit.

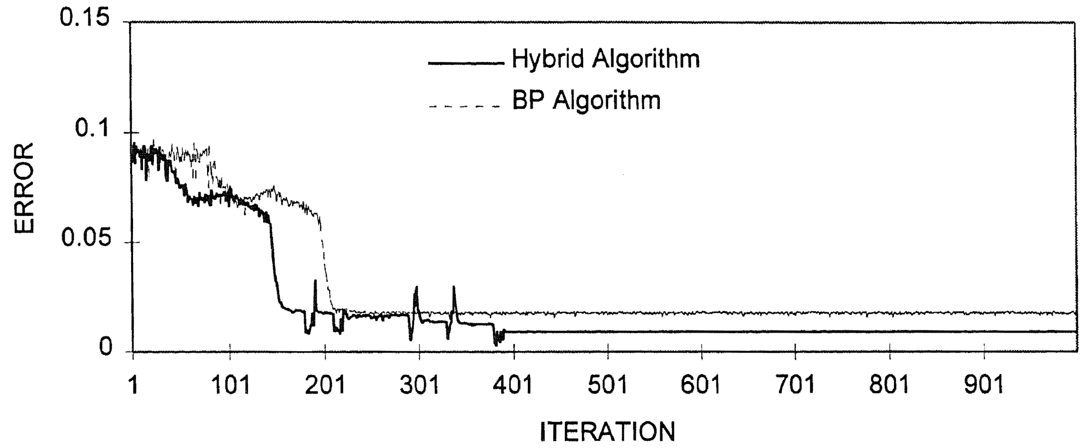


Figure 4.4: Learning curves of limited weight precision of 7-bit.

Table 4.1: Convergence Results of BP Method and Hybrid Method

		BP Method	Hybrid Method		
Weight Precision	Learning Rate	Conv. Rate	GT	ET	Conv. Rate
10-bit	0.95	0.8	0.001	0.01	0.8
9-bit	1.0	0.65	0.001	0.01	0.75
8-bit	1.85	0.7	0.001	0.01	0.9
7-bit	2.25	0.7	0.001	0.02	0.85
6-bit	2.25	0.7	0.001	0.02	0.85

It can be seen from the learning curves that the hybrid algorithm has the better capability to minimise the total error signal and gives faster convergence than the conventional standard BP method. However, it takes a relatively long time to settle when the weight resolution is low, due to the low precision causing a higher error signal. The convergence results show that the ability to converge at low weight precision of the proposed algorithm is higher than that of the conventional backpropagation method.

4.3.3.2 Overlapping Classes Classification Problem

An experiment was carried out in order to investigate the proposed algorithm for the problem where the output classes in the training set overlap. Both the standard backpropagation method and the proposed algorithm were used to train a network to separate two Gaussian noise sources. This benchmark problem was presented by Lovell and Bradley (1996) in their Multiscale Classifier evaluation. The first 1,000 points of the training set is shown on Figure 4.5. There was a training set of 2,000 examples and a test set of another 2,000 examples. The Multiscale Classifier with a pruned tree has achieved 94.9% accuracy on the test set of the noise source. Two networks both with 2 input nodes, 1 hidden node and 2 output nodes corresponding to each class were constructed. The first network was trained with the proposed algorithm and the second was trained with the standard backpropagation algorithm. All experiments were performed using the same initial random weight file. The Gaussian noise sources are normalised into 8-bits ranging between 0 and 1. After the training, the networks were tested with 2000 testing samples. If the output value of an output node exceeded the threshold (0.5) then it was considered to be 1, and if it was less than the threshold (0.5) then it was considered to be 0. The performances of the two networks on the test data set are summarised in Table 4.2.

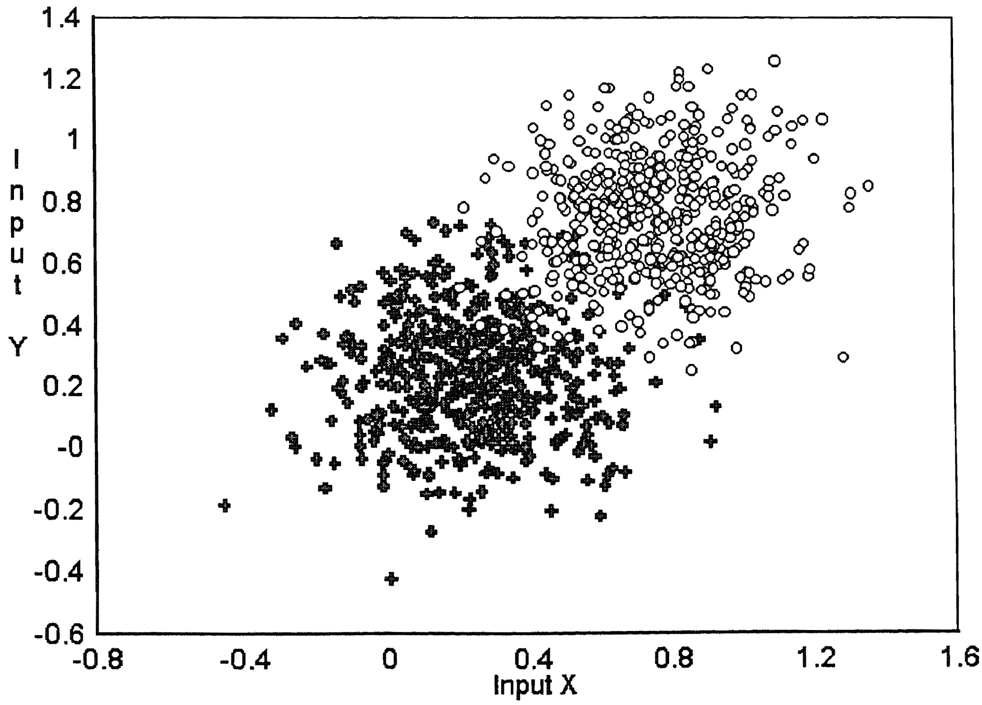


Figure 4.5: Overlapping Gaussian noise sources training data of first 1000 data points

The results given in Table 4.2 show that the network trained with the proposed algorithm at low resolution has a stable generalisation ability compared to the conventional backpropagation method. These results suggest that, if the training sample is large 5 enough, the proposed algorithm can learn quite well at low precision calculation.

Table 4.2: Results of BP Method and Hybrid Method on Gaussian Distributions of Test Set

		BP Method	Hybrid Method		
Weight Precision	Learning Rate	Performance	GT.	ET	Performance
8-bit	1.5	94.65%	0.0001	0.01	95.55%
7-bit	2.5	95.45%	0.001	0.02	95.9%
6-bit	3.0	95.2%	0.001	0.03	95.5
5-bit	4.0	94.2%	0.001	0.03	95.55%

4.4 The Selection of Low Precision Weights for BCG Classification

As discussed in chapter 2, there are two constraints in this BCG classification study. One is that the number of available samples is small and the other is that the distribution of heart attack patterns in the training samples is too small compared with other classes. If it is decided simply to rely on the hybrid learning algorithm for weight selection, it might be possible to miss the heart attack subjects because the small number of heart attack subjects can only make a small contribution to the final error function as we have discussed in chapter 3, section 3.4.3. This means that although the final error signal might be small, the misclassification of the heart attack subjects might be high. In a comparative study evaluating an approach that trained the network directly with fixed precision weights and a method that trained the network with high precision first and gradually reducing the weight precision, it was demonstrated that a better performance was achieved by using the second method (Yu et al., 1995). This is because, during the high precision training, the information can be properly coded into the weight connections. Therefore, in this experiment, the same strategy was adopted. The network was firstly trained with high precision calculations using the same training data set as was described in chapter 2, section 2.4.3.1.2. After the network converged, the hybrid algorithm was applied to train the network and to gradually limit the weight precision. The network was then tested using the testing data set to evaluate the performance.

As described in chapter 3, the network adopted in this study has 20 input nodes, 4 hidden nodes and 3 output nodes. Since the resolution of ADC for the BCG signal is 8-bit, it is considered that 8-bit precision is sufficient for the inputs to the neural classifier. The network was trained by updating the weights after every training sample. Consider the situation in which a training pattern p with a small cost function E_p is presented; it would be a waste of time to update the weights again for this pattern. Furthermore, this small error E_p usually has little effect on the MSE of E. Hunt and Deller (1995) have pointed out that it is almost unnecessary and computationally expensive to use all training data at every iteration to refine the weights. They gave the explanation as follows: "A training pattern whose individual performance surface is almost flat at the present weights does not contribute significantly to the determination of the gradient direction. The effect of including a training pattern with a small corresponding derivative, one whose performance surface is almost flat at the present weights, is smaller than the effect of another training pattern with a large derivation, or a performance error surface that is not as flat." Thus, concentrating on those patterns which make large contributions to the total error function and eliminating unnecessary weight updating procedure seems to be a way to speed up the training process (Sakaue et al., 1993). For the p th pattern, if all $|d_p - o_p| < T$, where T is a weight updating threshold value which is below 0.5, the update weight calculation is eliminated. Otherwise, the weights are updated using the learning rule. The overall training procedure used was as follows:

a) At first, the network was trained with high resolution calculation, using standard BP.
b) Set the GT and ET to appropriate values. Apply the hybrid learning algorithm to train the network, until the network converges.
c) Test the network performance with testing data set. If satisfactory, then go to (d). otherwise, exit.
d) Decrease the weights resolution by 1 bit and go to (b).

The training data set and testing data set used in this study were the same as described in the previous chapters 2 and 3. The 20 wavelet coefficients were normalised between 0 to 1 with 8-bit precision. The learning was initiated by setting the connection weights to random values between -0.5 to 0.5, and then it proceeded by interactively presenting the training data to the input nodes. The weight resolution was defined as 12-bit at first. A learning criterion of 0.01 for each pattern was set for the termination of the learning process. If the network had not converged after 1500 iterations, the training was stopped. Table 4.3 give the results of the network trained with 12-bit resolution.

Table 4.3: Classification Results of the Network with 12-bit Weight Precision

	Training Data			Testing Data		
Class	N	H	HA	N	H	HA
Normal (N)	30	0	0	25	2	1
Hypertension (H)	0	111		3	158	2
Heart Attack (HA)	0	1	2	0	1	2
Overall Performance		99.3%			95.39%	

By implementing the method of reducing the weight precision gradually, the minimum weight precision which could not affect the heart attack classes was found to be 6-bit. The results of variation of the weight precision are shown in Table 4.4 and Table 4.5.

Table 4.4: Classification Results of the Network with 6-bit Weight Precision

	Training Data			Testing Data		
Class	N	H	HA	N	H	HA
Normal (N)	30	0	0	26	2	1
Hypertension (H)	0	111	0	4	156	1
Heart Attack (HA)	0	1	2	0	1	2
Overall Performance		99.3%			95.36%	

Table 4.5: Classification Results of the Network with 5-bit Weight Precision

	Training Data			Testing Data		
Class	N	H	HA	N	H	HA
Normal (N)	30	0	0	26	1	1
Hypertension (H)	0	111	0	10	151	3
Heart Attack (HA)	0	2	1	0	1	2
Overall Performance		98.6%			91.8%	

4.5 Discussion

A hybrid learning algorithm has been developed to improve the MLP for fixed precision training. Two bench mark problems have been tested. Figure 4.1, Figure 4.2, and Figure 4.3 show the learning curves for the XOR benchmark problem of the hybrid learning algorithm and the standard backpropagation algorithm. The MSE curves show that the hybrid algorithm converges relatively quicker than the standard backpropagation algorithm. Although there were several oscillations which were caused by random optimisation, the MSE function can rapidly drop to a lower value. From the results of Table 4.1, it can be derived that the convergence rate of the hybrid algorithm is superior to that of the standard backpropagation algorithm. This suggests that the random optimisation method enables the network to escape the paralysis situation in low precision learning. The result showed in Figure 4.1 suggests that the choice of the number of iteration steps and threshold values has a significant effect on the network convergence. In the second benchmark problem, the hybrid learning algorithm was compared with the standard backpropagation algorithm on an overlapping classes classification. For the standard backpropagation algorithm, the network performance was tested with 7-bit to 6-bit weight precision. However, the generalisation performance was decreased when the weight precision was further reduced. The proposed hybrid algorithm had a consistent generalisation performance. These results demonstrated that a fast convergence ability and a robust classification performance can be achieved by the proposed hybrid algorithm compared to the standard backpropagation algorithm.

The proposed hybrid learning algorithm has been used to select the weight resolution for wavelet coefficients classification which was described in chapter 3. The training strategy proposed in a previous study was adopted. To improve the training speed, a further modification has been introduced to avoid unnecessary weight updates. By using the same partitioned data set, the effect of weight precision on the neural network performance for the BCG classifier was evaluated. Table 4.4 shows that when the weights were reduced to a 6-bit expressions, the results for the training data and the testing data set were the same as for the high precision training. When the weight precision was further reduced to 5-bit, the accuracy for the testing data set was decreased. This misclassification error is attributed to inadequate weight precision (Yoo and Pimmel, 1994). Thus, it is suggested that 6-bit is the minimum weight precision to keep the required performance with the available data set.

Research has shown that the performance of feedforward neural networks depends on both weight precision and the number of hidden neurons (Nakayama et al., 1990; Sakaue et al., 1993). The weight precision could be further reduced below 6-bit while keeping the same performance, if extra hidden neurons were added. However, as discussed in the previous chapters, the available data set composition constrains the evaluation of network generalisation in this study. On the other hand, adding an extra neuron means that extra I/O (input/output) resources are required. Thus, in this study, the minimum number of hidden neurons are used in a trade-off between circuit complexity and I/O resources.

4.6 Conclusions

In this chapter, a hybrid learning algorithm has been described to improve the network learning ability in low precision training. Two benchmark examples commonly used in the literature are used to compare the effectiveness of the proposed hybrid algorithm and the standard backpropagation rule. From the experimental results, it is suggested that the new hybrid learning algorithm has good generalisation and convergence capability in discrete weights learning compared to the conventional backpropagation method. It was noted that in limited precision learning, the convergence of the standard backpropagation is sensitive to the initial weight values. However, the proposed algorithm offers robust performance. Finally, the hybrid algorithm is applied to BCG classifier weight precision selection. A training strategy is developed using high precision first and then reducing the weight precision gradually. The result demonstrates that the generalisation capability of a MLP with 6-bit precision weight provides comparable performance to the high precision network.

The advantage of the proposed hybrid learning algorithm is that the additional calculations involved for the modification are relatively small. It offers high speed of convergence and robustness compared with conventional backpropagation rule. This feature makes the hybrid algorithm applicable for large size networks training to save time. Another advantage of the hybrid algorithm is that it is possible to allow the network to escape from regions of the weight space from which the standard gradient descent might need a lot of iteration to escape. In this study, the proposed hybrid algorithm is compared with the standard backpropagation algorithm. However, a further investigation is required to compare the proposed hybrid algorithm with other modified BP methods (e.g. Dundar and Rose, 1995; Alibeik et al., 1995) to justify the effectness of the proposed hybrid algorithm.

Chapter 5

Method of a Fully Parallel Neural Network Implementation in FPGAs

5.1 Introduction

In chapter 3, an automatic classification system based on the wavelet transform and feedforward neural network has been proposed. Furthermore, a method to reduce the weight precision has been introduced in chapter 4 to simplify the hardware implementation for real time application. As discussed in chapter 1, one of the primary objectives of this project is to explore parallelism in hardware using commercially available VLSI chips. Such an advancement will create the potential for new biomedical applications for diagnostic, monitoring, and life support systems. For instance, a fast device is required to deal with three channel vectorballistocardiography signal processing in real time applications (Schwerdt et al., 1987). The expanding frontiers of vision technology in biomedical applications such as computerised tomography (CT) and magnetic resonance imaging (MRI) demand even more computation power to process high dimension data (Christensen et al., 1996). Increasing demands of speed and higher performance have made research in the hardware implementation of algorithms in the fields of pattern recognition and machine intelligence very active in the last decade.

The common characteristic of neural networks or neural-like algorithms is that they are parallel in nature and can employ large amounts of computation in parallel. These requirements have presented significant challenges in parallel computing technology and very large scale integration (VLSI) architecture research. Due to the fast progress of VLSI technology and its trend in cost reduction, it possesses the high speed and massive computing capabilities which provide effective and economical solutions in real world applications.

There are a wide variety of implementation techniques in parallel network studies. Many researches have illustrated that the approach for improving performance in a specific application is to implement the specific functions and algorithms into specific integrated circuits (Hammerstrom, 1990; Boser et al., 1992). In chapter 4, the author has discussed the off-line learning method to reduce the weight precision and the simulation results have shown that 6-bit weight precision is adequate for this particular application. This result suggests that research into special-purpose hardware allows optimisation at every level of the implementation and permits certain functions to be realised in much smaller areas and to operate much faster than standard hardware. Selection of the integrated circuit technology is a problem dependent decision. The cost, performance, flexibility and schedule for the product would all be affected by this decision. In most medical applications, data collection is a costly and time consuming part of the work. Flexibility and performance are usually the main considerations which determine whether a design can accommodate later data expansion. On the other hand, reducing the time to market is one of the great challenges faced in the instrumentation industry today.

With the advance of large look-up table based FPGAs, the necessary performance and density are now available in a programmable form. High density FPGAs allow the designer to implement complex circuits on a single chip. Any of a number of potential configuration programs can be downloaded into the system's FPGAs to alter the logic or operation for particular applications as needed. There are proposals for adopting FPGAs for rapid in house prototyping (Noore, 1991), the use of FPGA based reconfigurable systems for testing designs (Bout et. al., 1992; Farrow, 1987), and the use of FPGAs as an alternative solution to 'hardwire' DSP algorithms in digital logic (New, 1995; Altera, 1996). Thus, it makes sense to use FPGAs for portable system prototyping, to respond to new parameters, application conditions, or special customer requests. This can significantly reduce the hardware resources and the design cycle. Furthermore, as the price of FPGAs rapidly declines, using FPGA technology provides an effective and low cost method for the development and prototyping of applications on hardware platforms.

Implementing a parallel neural network on the FPGA devices have been reported (Cox and Blanz, 1992; Botros and Abdul-Aziz, 1993). However, the reported works based on the XC3000 series needs tow chips for one neuron design due to the limitation of the architecture and logic density of XC3000 series. This makes the network bulky. The new XC4000 series offers the different structure and high logic density. Therefore, the aim of this study is to exploit the one neuron one chip solution by using improved FPGA architecture for specific applications.

This chapter is organised as follows: In section 2, following a brief discussion of technique and methodology in hardware implementation of neural networks, FPGAs are considered for the hardware platform. In section 3, following a survey of related work on recent developments in neural hardware implementation with FPGAs, a method is described for implementing a fully parallel neural network on the Xilinx 4000 family chips by making use of the fast carry logic feature. The design method is finally summarised in section 4.

5.2 Technology Selection for Neural Network Implementation

The techniques of neural hardware implementation are various and also are size dependent. Analogue circuits are quite attractive in terms of hardware size, power consumption, and speed. However, they suffer the drawbacks of sensitivity to noise, lack of scalability and difficulty in storing moderate weight precision. Digital circuits offer greater flexibility, scalability, and accuracy, and lower power consumption than analogue circuits. However, the disadvantage of the digital circuit approach is that a large silicon area is required to implement multiplication. A combined analogue and digital circuit approach has been used to trade-off these problems (Lee et al., 1993). Research into an optoelectronics approach has also been conducted for large size networks which are massively parallel. Although optoelectronic, analogue and hybrid implementations are not rare, usually the underlying technology is digital. This study focuses on the digital implementation technique.

Tremendous effort on neural hardware studies has produced various implementations over past decades. The model orientated implementation, such as large feedforward neural networks (Yasunaga, et al., 1993), Kohonen maps (Onodera and Takeshita, 1993) and cellular neural networks (Miyanaga and Tochinai, 1993), aims at proof of concept. These researches have produced large systems, but these do not migrate to the field easily because of their high cost (Moini et al., 1995). In most of the specific applications, the special purpose systems have a fixed network topology and fixed weights. In these cases, networks are often fully implemented by a special piece of hardware. Custom application specific integrated circuits (ASICs) provide an approach to implement neural like algorithms into hardware with the benefits of wide availability and higher performance through taking advantage of parallelism and optimised circuit placement (Boser et al., 1992; Watanabe et al., 1993). However, the costs, in terms of time and fabrication charges, of building application specific circuit systems are high. It is necessary to perform enough simulations before committing the design to silicon (Gillis et al., 1996). Furthermore, the hardwired design lacks flexibility in the situation that a late design change in functions or algorithms is required. ASIC technology makes sense in high volume applications from both cost and performance standpoints (Chen and Rabaey, 1992). It rarely makes a cost-effective solution for low-volume products, or at the system prototyping stage.

On the other hand, most off-the-shelf hardware chips that implement, in a general way, a given neural model usually are inefficient in circuit size, flexibility, speed and power consumption. Moreover, in the spectrum of application-orientated neural circuits research, there is little market for moderately parallel, general purpose neural chips. There is a niche of small-to-medium scale neural circuits that are tailored to the specific needs of given real time applications. For most of the specific purpose applications, the weight precision can be trimmed to lower precision after the proper training. In such situations, as the custom chips have a fixed arithmetic precision, the low precision calculation might waste the extra resources of the standard device. On the other hand, research to develop both the models and their optimal implementation is still limited (Hurdle et al., 1993). Meanwhile, the need for sophisticated algorithms for processing, coding, and automatically interpreting the information of biomedical signals is increasing. Therefore, using programmable techniques for study is sensible.

In the off the shelf VLSI digital hardware domain, FPGAs provide a popular alternative to standard chips and mask ASICs. High density FPGAs and associated complex programmable devices (CPLDs) share the ability of ASICs to fit the architecture to the algorithm. One benefit of the FPGA technology is the flexibility that allows the circuit to be reconfigured in system, which is ideal for adaptive system design. With such technology, the function in the hardware can be reprogrammed and loaded with new algorithms under program control.

It is considered that FPGAs offer an ideal platform for prototyping specific circuits in medical applications. First, to develop truly robust ANNs, investigators are required to train their networks on huge training data sets. However, in real world application, it is not realistic to obtain large data samples at the initial phase. In this application, for example, the data samples for heart attack subjects take several years to gather. Thus, when the patient database expands later, the neural network parameters (i.e. weights) or even the network configuration needs to be changed to improve the diagnostic accuracy. The reprogrammability of FPGAs in system is suitable for this dynamic environment. The low price and high programmability of the FPGAs offers an alternative to solve this problem. Second, recent advances in semiconductor technology and improved architecture in FPGAs has made them available in a complexity which is suitable for complex circuit design, with many mature and powerful CAD tools to support the FPGA development. The development of logic synthesis tools have allowed the hardware designs to be described in high level hardware description languages (HDL). This makes the hardware design on the FPGAs more flexible.

5.3 Implementing Neural Networks with Xilinx FPGAs

5.3.1 Reconfigurable Logic Technology

Current commercial FPGA products can be categorised into static random access memory (SRAM) based FPGAs and antifuse-based FPGAs. The SRAM based FPGA has programmable connections controlled by SRAM cells and basic lookup logic blocks. In these FPGAs, the logic functions and interconnections are determined by configuration program data stored in the internal static memory. Thus, they can be re-configured at any time in the application. The advantage of SRAM based FPGAs is that the hardware design can be adapted for varying tasks. The main manufacturers of SRAM based FPGA are Xilinx, Altera, AT&T and Atmel. The similarities and differences in function and architecture of these products can be found in Brown and Rose (1996).

The anti-fuse based FPGAs can be described as one time programmable devices. The connections between logical cells can be programmed by the user. Once the circuit is verified, a special high voltage is applied to fuse the connection permanently. Compared with the SRAM based FPGAs, the anti-fuse based FPGAs are one time programmable devices that cannot be reconfigured in circuit.

However, the antifuse switching elements are much smaller than the RAM-based switches, allowing more routing resources with less silicon area. They also provide design security. The leading manufacturers for such devices are Actel, Quicklogic and Cypress. Brown et al. (1992) have compared the details of FPGAs and CPLD architectures and their functions.

Although there are many types of FPGA architectures and development tools, the Xilinx development system and the Viewlogic CAD (computer added design) tool are used in this design because there were existing education versions at the Faculty of Design and Technology , the University of Luton, and to meet the financial constraint in this project.

5.3.2 Previous Work on Neural Network Implementation with FPGAs

There is a growing interest in FPGA based neural network implementation. Cox and Blanz (1992) reported the implementation of a feedforward network in lookup table based FPGAs for multiple pattern recognition tasks without a physical change in the system. The fully interconnected one hidden layer feedforward neural network with 12 inputs, 14 hidden units and 4 outputs was implemented with Xilinx FPGAs. The resolution of all values was 8-bit. Each neuron occupied two XC3090 FPGAs and the sigmoid functions were implemented in programmable read only memory (PROM). The output of each neuron was used as address data for the PROM table. The weights were hardwired into the multiplier logic. In order to accommodate the Xilinx architecture, each 8-bit by 8-bit multiplication was split into two 8-bit by 4-bit multiplications with one final addition operation to achieve the goal of minimum logic area on FPGA. A combined three to two and carry save adder tree was used to achieve the final summation in parallel. The implemented classifier achieved 4.48 billion connections per second (BCPS) performance. The authors have applied the same classifier to several industry image processing problems. They demonstrate the advantage of reconfigurable hardware in that it allows users to reconfigure the weight parameters for different image processing applications without physical change. However, the work shows that one neuron implementation needs two chips, even though the maximum density XC3090 is used. Thus, this makes the network design bulky.

Botros and Abdul-Aziz (1993) have reported a neural network with 5 inputs, 4 hidden units, and 2 outputs implemented with Xilinx FPGAs. The weight multiplier logic implementation was similar to the work reported by Cox and Blanz (1992). The sum operation was implemented on a combined carry look-ahead adder and an accumulator. The final output of the accumulator was used to address a sigmoid function look-up table which was implemented in an EPROM. Each neuron consists two XC3042 FPGAs and one 1K×8 EPROM. For the 5 inputs, 4 hidden units, and 2 outputs network, the simulation results showed that 4 million connections per second (MCPS) was achieved. The drawback of this design is that the inputs to the hidden neurons are sequential through a multiplexer, although hidden neurons processing is parallel. Thus, the system throughput depends on the number of inputs so that the system performance will degrade when implementing a middle size network. Furthermore, two chips are required for one neuron design, even though the network is small. Compared with the work of Cox and Blanz (1992), the design for final result summations is simplified. However, the performance and compactness of the design is not as good as Cox and Blanz's implementation.

Considering the expense of digital implementation of multiplication, several researches were reported to implement non-multiplication neural models on FPGAs. Hikawa (1995) demonstrated the implementation of an on-chip learning multiplier-less multilayer neural network proposed by Marchesi et al. (1993) with FPGAs. In this method, the sigmoid function was replaced with a comparator and all calculations are replaced with simple summation. To further reduce the circuit's size, bit serial communication and bit serial arithmetic operation were used. A three input, four hidden nodes and two output nodes network is implemented on a Xilinx FPGA (XC4013). The simulation shows that 14.6 MCPS was achieved. This work has provided a potential application of implementing a on-chip learning neural network on FPGAs to achieve high performance for real time application. However, in most practical applications, the neural models to be implemented require the multiplication operations.

Several researches have been reported to improve the use of the circuit density of FPGAs for large neural network implementation. Eldredge and Hutchings (1994) explored the run-time reconfiguration property of FPGAs to implement neural co-processors with on-line learning ability. The neuron processing was divided into three sequential execution stages: feed-forward, backpropagation, and weight update. At each stage the hardware circuit module was configured on the FPGA once during the process. The advantage of this method is that it improves the hardware utilisation, so allowing more hardware neurons to be implemented on the available FPGAs resources. Thus, the hardware density required can be greatly reduced. This work demonstrates a framework of implementing neural network using run time reconfigurability of FPGA to save hardware resources. However, this method cannot be used for high speed applications since reconfiguration time is of the order of milliseconds. The design has been implemented on the XC3090 Xilinx product and simulated at 14 MHz clock. The results show that the network performance increases monotonically with the number of FPGAs used. For 10 FPGAs, the performance was about 10 MCPS.

Bade and Hutchings (1994) investigated the feasibility of using FPGAs to implement the stochastic neural network architecture. The architecture was developed specifically for Xilinx FPGAs. All the calculations and activation functions are mapped into the lookup tables. A single layer neural network with 12 inputs and 10 outputs was implemented on one XC4003, which has 100 CLBs. It is estimated that this architecture could operate at 21.7 MCPS. Although this method can significantly reduce the circuits required for neural network implementation, the control complexity is increased as the bit serial stochastic computing needs precise timing control.

The implementation of neural networks with FPGA can be roughly divided into two categories, serial and parallel. Serial design allows more compact implementation with multi clock cycles and the parallel design allows the network implementation performance complex operation in a single clock cycle with expense in circuit area. For this BCG classifier implementation, both serial and parallel design can be adopted because the signal sampling rate is slow. However, motivated by the potential requirements of parallel processing in the biomedical field. This case study focuses on neural network parallel implementation with FPGAs.

Although some early work has been done on parallel implementation of parallel neural networks, most implementations using the XC3000 series need complex arithmetic unit logic due to the cumbersome nature of XC3000 architecture. One neuron design usually needs two FPGA chips (Cox and Blanz, 1992; Botros and Abdul-Aziz, 1994). As there is an on going FPGAs technology curve that continues to provide high density logic and improved architecture, it would be desirable to exploit the improved architecture of FPGA series in implementing arithmetic units to produce a more compact neuron design. As the Xilinix 4000 series offers the improved architecture and high density, it is a goal of this study to exploit the effect on the hardware size and performance for a parallel network implementation. The available CAD development tools and Xilinx XC4000 library resources are used in this study.

5.3.3 Design Considerations

The network to be implemented has 20 inputs, 4 hidden nodes and three output nodes as described in chapter 3. Using the simulation results of chapter 4, the weight precision is 6-bit and the input precision is 8-bit. It has been shown in Chapter 2 that for the classification phase, the particular implementation of each neuron simply performs the operation of multiplying each input with its weight and summing the products. The final result goes through a sigmoid function. For the input neurons, they just simply buffer the input data. In order to achieve the maximum performance, it is most desirable to exploit the maximum amount of parallelism. This study focuses primarily on the parallel implementation of a neural network with efficient logic area on XC4000 FPGAs.

5.3.3.1 Multiplication

The digital multiplier design provides the key to neural network performance and complexity. Typical high performance multipliers employing Booth encoding (Huang et al., 1994), and Wallace tree (Nichani and Ranganathan, 1992) techniques, would require complicated interconnection patterns which are expensive to route on an FPGA. The carry save adder (CSA) array technology is the most popular method for parallel multiplier implementation (Lee et al., 1987; Lu and Samueli, 1993). However, CSA array multiplier needs a large logic area in an FPGA. There are several types of multiplier design that make use of the properties of look-up table based FPGAs. These include the ROM (read only memory) based multiplier, the recursive add-accumulator multiplier, logic tree multiplier and small look-up table multiplier.

As each function generator of the XC4000 can be configured as on chip ROM, the products can be precalculated for all possible inputs and stored in the ROM. In the ROM multiplier, all possible products are precalculated and stored in the ROM. The inputs then are used to address the ROM. The product is read back as data from the ROM. However, the ROM-based design could need large FPGA resources when large inputs are involved. For this application, one 5-bit weight by 8-bit multiplication requires a 13-bit wide and 256 words deep (ROM) lookup table which needs 147 CLBs. One hidden neuron has 21 weight multiplications which means that 3,087 CLBs are required. This would need several FPGA chips to meet the requirements.

The recursive add-accumulate multiplier is an area efficient implementation and requires a clocked system. In this design, each bit of the multiplier factor is used to do an add/pass operation; the shift is also done in routing. A single add/pass stage is used with a single accumulation and the partial product held in the accumulator is fed back to the add/pass stage each clock cycle. The output of the add/pass stage is routed to the next clock cycle of the multiplication to provide the product. Although area-efficient in design, this multiplier has the constraint that it can only execute one bit of the multiplier operation per clock cycle. For each pair of factors, a product will be available in the number of clock cycles equal to the number of bits in the multiplier factor. Thus, the performance of this design is limited.

In a logic tree multiplier design, each of the resultant bits is a logic function of the relevant bits of each operand. The logic involved tends to have a high fan-in requirement and the gate count depends on the input resolutions. As each CLB has maximum inputs of 9, large inputs must be split to fit the architecture. This prevents a compact design and also increases the design complexity.

Cox and Blanz (1992) have implemented implement dedicated arithmetic by 'hardwiring' the constant weights in the FPGA logic for multiplier design. The basic idea is that the weight value is known prior to the hardware design. Thus, each of the result bits can be expressed as a function of the input variable. As each CLB can perform two independent logic functions on any 4 inputs, an input variable that is more than 4-bit can be split into several 4-bit input to generate the partial products. These partial products can be added in a final stage to form the multiplication product. Cox and Blanz (1992) have shown that this method is an efficient way of saving logic for multiplication implementation. This method has been adopted in this study. However, instead of keeping the lower partial product positive to eliminate the need for sign extension when adding it to the upper partial product, a two's complement multiplier is implemented (Yu and Dent, 1994). In this application, the multiplication is a 6-bit constant multiplied by an 8-bit input. Thus, multiplication was split into two 6-bit by 4-bit multiplications with a final adder.

The weight multiplication consists of two parts: generation of partial products and addition of partial products. Let the $W[5..0]$ represent the constant weight and $X[7..0]$ represents the input variable. The two partial products can be expressed as follows:

$$P = W[5..0] \times X[3..0] \qquad (5.1)$$

$$P' = W[5..0] \times X[7..4] \qquad (5.2)$$

where the P represents the low partial products and P' represents the high partial products.

The Baugh and Wooley (1973) two's complement algorithm is used for direct two's complement multiplication. The advantage of this algorithm is that the negative numbers are transformed, allowing direct addition. As the input value is always positive (sigmoid function), the partial product is generated as shown in Figure 5.1. The whole product value can be obtained by adding the two partial products together as shown in Figure 5.2. It is not possible for the adder to overflow because it is known that multiplication of two integers of 6-bits and 8-bits respectively will generate a 6+8 bit result.

$$
\begin{array}{cccccccccc}
 & & & & w_5 & w_4 & w_3 & w_2 & w_1 & w_0 \\
 & & & & & & x_3 & x_2 & x_1 & x_0 \\
\hline
 & & & & & w_4x_0 & w_3x_0 & w_2x_0 & w_1x_0 & w_0x_0 \\
 & & & & w_4x_1 & w_3x_1 & w_2x_1 & w_1x_1 & w_0x_1 & \\
 & & & w_4x_2 & w_3x_2 & w_2x_2 & w_1x_2 & w_0x_2 & & \\
 & & w_4x_3 & w_3x_3 & w_2x_3 & w_1x_3 & w_0x_3 & & & \\
1 & w_5\bar{x}_3 & w_5\bar{x}_2 & w_5\bar{x}_1 & w_5\bar{x}_0 & & & & & \\
\bar{w}_5 & & & & w_5 & & & & & \\
\hline
p_9 & p_8 & p_7 & p_6 & p_5 & p_4 & p_3 & p_2 & p_1 & p_0
\end{array}
$$

Figure 5.1: 6-bit by 4-bit Baugh-Wooley Two's Complement Multiplication.

$$
\begin{array}{cccccccccccccc}
 & & & & & P_9 & P_8 & P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0 \\
+ & P'_9 & P'_8 & P'_7 & P'_6 & P'_5 & P'_4 & P'_3 & P'_2 & P'_1 & P'_0 & & & & \\
\hline
 & O_{13} & O_{12} & O_{11} & O_{10} & O_9 & O_8 & O_7 & O_6 & O_5 & O_4 & O_3 & O_2 & O_1 & O_0
\end{array}
$$

Figure 5.2: 6-bit by 8-bit parallel multiplier implementation

5.3.3.2 Summation

The next step for each neuron is to sum all products together to form a final result. Among the solutions proposed in the literature, carry lookahead adders (CLA) are the fastest (Hwang, 1979). However, they require a large numbers of CLBs. The carry-save adder (CSA) tree (Yu et al., 1995), or the combined three to two adder and carry-save adder tree (Cox and Blanz, 1992) have a compact layout on FPGA and can have a fast throughput by incorporating a pipelining design. For XC4000 architecture, one CSA needs a single CLB. However, for the ripple carry adder (RCA) design, each CLB can implement 2 ripple carry adders which means that it uses the least amount of logic area. Furthermore, due to the fact that dedicated carry circuitry in XC4000 device is so fast and efficient, the conventional speed-up methods using additional logic to generate the fast carry are meaningless compared with the ripple carry adder for 16-bit wide designs (Xilinx, 1994). The ripple carry adder has an easy and compact layout on the XC4000 series FPGA and has no delay penalty, if the adder size is not large. Thus, the ripple carry adders are adopted in the design as an area efficient addition architecture. For the hidden layer, each node sums 20 weight multiplication products plus a bias. The 21 additions are realised with a ripple carry adder tree to produce the final summation.

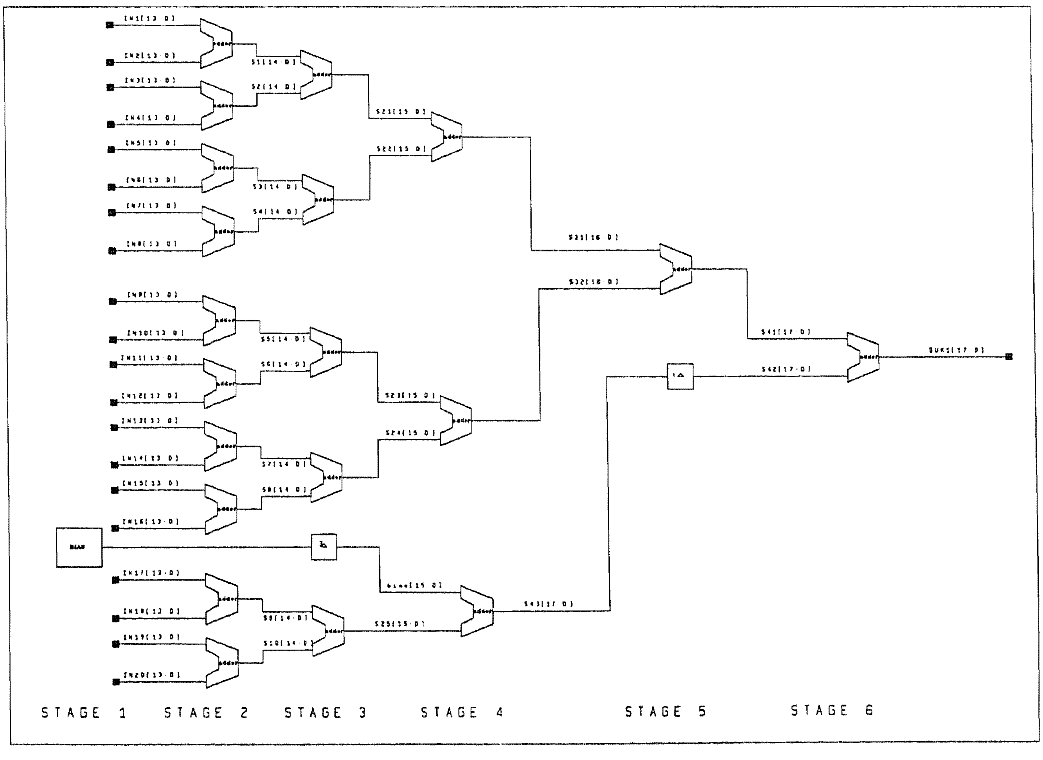


Figure 5.3: The pipelined adder tree used in the hidden neurones.

In this application, only a high throughput is required and constraints on latency are not a critical issue. Therefore, it is convenient to adopt a pipelined architecture. Because the data flow in the adder tree is along a path without any feedback loop, the system can be set to work on a higher clock rate by breaking the path and inserting pipeline registers. In the previous section, it was mentioned that each CLB has two flip-flops available "for free". By making use of these two flip-flops as registers for two function generator outputs, the pipeline design can be completed without any logic expense. The summation adder trees implemented in one of the hidden nodes consist of six levels each with a pipeline register as shown in Figure 5.3. The first pipeline stage corresponds to the output of the partial products adder of the weight multiplier. The 14-bit products are latched in the first 14-bit pipeline register as the summation adders inputs. Here, the maximum time delay is determined by the longest ripple carry adder. In the hidden layer neurones, the maximum adder delay is dependent on a 18-bit ripple carry adder. For the output layer, each node only has 4 feature inputs. Thus the adder trees are simpler than in the hidden nodes, as will be discussed in the next section.

In real time application, it is important to prevent overflow to keep the result right. One solution is to use the sign extension technique before adding two integers together. However, this method will need an extra ripple carry adder stage. It will be more convenient to use the overflow and final sign extension to generate the sign extension bit. There are three possible cases in two's complement addition (Hwang, 1979). Let A and B be two's complement integers with n bits. In case one, both A and B are positive. No carry can be generated out of the sign position, if $A+B<2^{n-1}$. Thus, the addition result $S=A+B=s_{n-1}\dots s_1 s_0$ is in correct form and the sign bit s_{n-1} will be zero (positive). If $A+B\geq 2^{n-1}$, the carry will enter the sign position and overflow will take place.

In case two, both A and B are negative. The two's complement of these two integers can be expressed as $\widetilde{A}=2^n-A$ and $\widetilde{B}=2^n-B$. In this case, both a_{n-1} and b_{n-1} are always "1". There is always a carry out of the sign position and this carry needs to be ignored to keep the right result. The result is in correct form $2^n-(A+B)$, if $A+B\leq 2^{n-1}$. Otherwise, an addition overflow has occurred if $A+B>2^{n-1}$.

In case three, one of the integers is positive and the other is negative. In this case, no overflow will occur. The result will always be in correct form. Considering the above three cases, an extra bit is added to the ripple adder to prevent the overflow cases. Table 5.1 summarises the logic truth table of the above three cases, where c_{n-1} represents the carry generated by sign bit s_{n-1}, the c_{n-2} denotes the carry entering into the sign position, and the s_n is the extended bit which is used to prevent the overflow. In the simulation study, it was shown that no overflow occurs for the final adder in the adder tree. Thus, no extend bit is needed for the final adder implementation.

Table 5.1: Logic Truth Table of Adding Extra Bit to Prevent the Overflow

c_{n-1}	c_{n-2}	s_{n-1}	Overflow	s_n
1	0	0	1	1
1	1	1	0	1
0	1	1	1	1
0	0	0	0	0
1	1	0	0	0
0	0	1	0	1

By simplifying the Table 5.2 logic truth table, one can get an adder with an extra bit in the final sum result. An adder tree based on the overflow prevention adder was constructed to generate the final sum.

5.3.3.3 Activation Functions Implementation

In the basic computation of the feedforward neural networks, the final accumulated result is evaluated by the activation function which in our application is the sigmoid function $f(o) = 1/(1+e^{-o})$. Because of the computational complexity of implementing the sigmoid function, the efficient implementation in digital design is to use a lookup table instead of direct calculation. This look-up table can easily be implemented on the off-the-shelf programmable read only memory (PROM) or erasable programmable read only memory (EPROM). The output of the accumulated result is in 18-bit two's complement form. However, an 8-bit two's complement result is required to be output by the hidden neurons. This can be achieved by using the scaled accumulated results to address an 8-bit wide look-up table. By further looking at the sigmoid function behaviour in Figure 5.4, it is shown that the output of the sigmoid function saturates to approximately 1.0 when the summation result $o \geq +8.0$ and saturates to approximately 0.0 if the summation result $o \leq -8.0$. Therefore, 9-bit plus one sign bit is adequate to address the sigmoid look-up table. Accordingly, 18-bit accumulated result is scaled to address a look up table. If the value is greater than 8.0 or less than -8.0, a saturation flag is set. The sign bit is added to the remaining 9-bit value and a saturation flag to address the sigmoid function look-up table.

For the output layer nodes, one is only interested in the class status rather than actual output values. Thus, it is not necessary to let the output summation go through the ROM lookup table. The final summation results can compete using the winner take all rule which was discussed in chapter 2. The magnitude comparators are implemented to compare the magnitudes of two numbers. Again by using the fast carry logic, 2-bit subtraction operations can be implemented in a single CLB. The outputs of the comparators are 'Great Than (GT)' or 'Less Than (LT)', if the two number are different. If the two numbers are the same, both GT and LT are set to logic '0'. Figure 5.5 shows function diagram of the three output layer nodes. Two nodes are compared first. The comparison result controls a multiplexer which allows the bigger magnitude number to go through. If these two number are the same, one of the numbers will go through. The output result of the multiplexer is compared with the third number. The comparison results drive a logic expression function. The largest magnitude number will set to logic '1' and the others will set to logic '0'. If the three numbers are the same, then the three outputs will all be set to logic '0'.

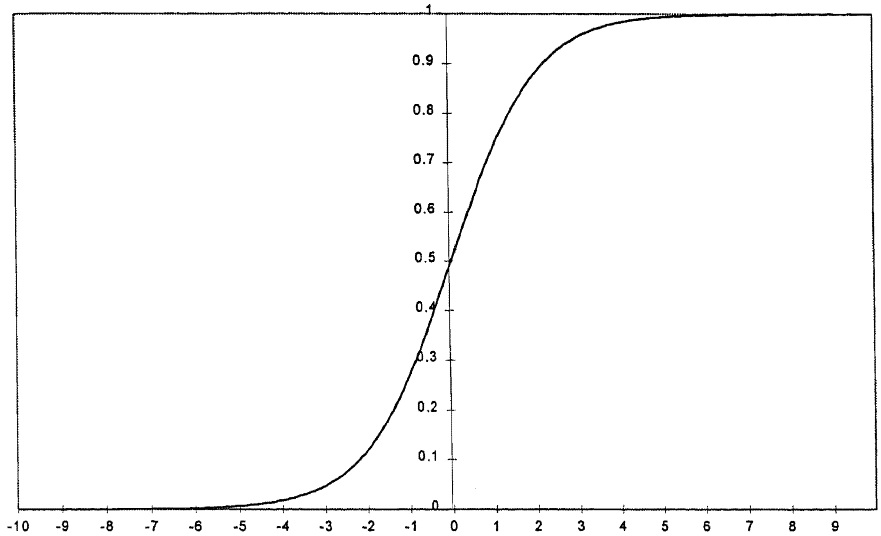


Figure 5.4: Sigmoid function

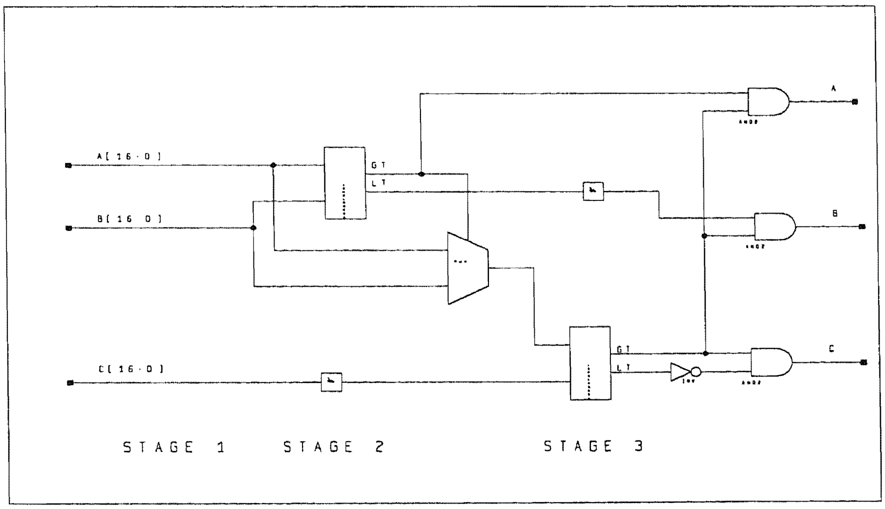


Figure 5.5: Function diagram of implementation of the three output neurones in output layer

5.4 Summary

In this chapter, the use of SRAM based FPGA technology for the digital implementation of neural networks has been discussed. It is shown that FPGA technology allows users to implement optimised designs to achieve the required performance. Because of its reprogrammability, it is useful in a range of biomedical signal processing situation where the algorithm parameters need to be updated as the database expands. It is argued that the FPGAs/CPLDs offer ease of use, off-the-shelf availability, reprogrammability, and low cost features for medical instrument prototype studies.

Although early works have addressed parallel implementation neural networks with SRAM based device (Cox and Blanz, 1992; Botros and Abdul-Aziz, 1993), the design issue is associated with the XC3000 series structure. In these designs, one neuron implementation needs two chips which makes the network bulky. As the new family of XC4000 offers an improved architecture, a continuing investigation into implementing neural networks on improved architecture FPGAs to achieve one neuron one chip solution is still necessary. In this study, a method of implementing neurons for a fully parallel network design on Xilinx XC4000 series FPGAs has been described. The network has 20 inputs, 4 hidden nodes and 3 output nodes. The activation function for the hidden nodes is the sigmoid function. For the output layer nodes, the winner take all rule which was discussed in chapter 2 is adopted. The hardwired weight multiplication scheme is similar to the previous works (Cox and Blanz, 1992; Botros and Abdul-Aziz, 1993). To make use of the feature of XC4000 architecture, a parallel ripple carry adder tree is adopted. The pipelined stages have been incorporated in the RCA trees to achieve a high throughput. It is shown that the CLBs required for implementing adders for XC4000 series are less than XC3000 series.

Chapter 6

Implementing Neurons in FPGAs

6.1 Introduction: Overview of Xilinx FPGAs Design

Based on the discussion of the methodology of implementing arithmetic units on FPGAs in chapter 5, this chapter describes the experiment of implementing the neuron function on Xilinx FPGAs for real time application. The main goal is to show that by using the available development tools and resources, one could develop a practical system with adequate performance and flexibility. The implementation procedure consists of two stages; individual neuron design and connecting these neurons together to construct a fully parallel feedforward neural network. However, resources were not available to transfer the design into hardware. Only the first stage is carried out in this study.

The development software used is Workview PLUS version 5.2 and Xilinx XACT version 5.1 which can only implement devices up to 13, 000 usable gates (corresponding to the XC4013 FPGA). All designs were conducted on a 486-33 PC with 32 mega-byte RAM.

The complete design flow with Viewlogic and Xilinx XACT is shown in Figure 6.1. The custom logic is entered either using a hardware description language (HDL) such as Very High Speed Integrated Circuit HDL (VHDL), or through the use of the ViewDraw schematic capture tool and the XACT logic library (Viewlogic, 1993). The completed design is synthesised with the Xilinx library. This automatically generates a Xilinx Netlist Format (XNF) file. The netlist file can be used for logic function simulation. After verifying the functionality of the complete design from the ViewSim simulation, the XACT software is applied to partition and place the design into logic blocks with a near-optimal placement for each block, and finally to select the interconnect routing to fit the XC4000 architecture. Once the design is placed and routed, additional timing information due to the actual interconnection is back annotated to the Xilinx netlist and can be used for static timing analysis. Once the procedure is finished, the completed design can be translated to a serial bitstream file to be down loaded into the FPGA hardware.

The organisation of this chapter is as follows. Section 2 describes the detailed design of the multiplier and the ripple carry adders. The neuron chip layout and test results are presented in section 3. Section 4 discusses the implementation, performance and flexibility. The conclusions are given in section 5.

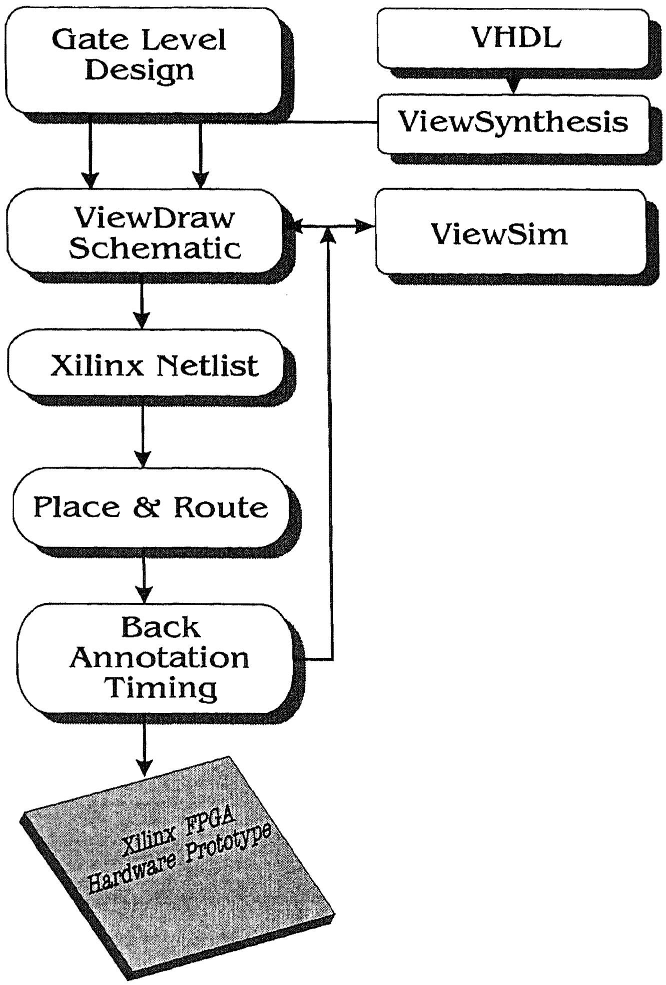


Figure 6.1: Xilinx FPGA design flow

6.2 Neuron Implementation

A combined VHDL and schematic design entry is adopted in this study. This is because the fast carry logic implementation imposes restrictions on using the soft macro library. Macros are easily integrated with schematic design entry and can be customised to meet individual design requirements, for example, adders, subtractors, or counters. The design for a 6-bit by 4-bit Baugh-Wooley two's complement multiplication described in section 5.3.3.1 of chapter 5 is entered in VHDL code. The available Viewlogic synthesis tool has problems in removing redundant logic during the synthesis process. This makes it inefficient for large logic circuit synthesis. In order to make sure that one partial product is mapped into one function generator of a CLB, each partial product generator is defined as an individual unit and the partial multiplication is expressed as Boolean logic. An example generating partial product bit 4 is as follows:

```
library synth;
use synth.stdsynth.all;

ENTITY product4 is
        PORT( SIGNAL feature : IN VLBIT_1D (3 downto 0);  -- 4-bit inputs
            SIGNAL pl_4 : OUT VLBIT    -- 1-bit output product
          );
        END product4;

ARCHITECTURE multiplication OF product4 IS
  CONSTANT weight : VLBIT_1D (5 downto 0) := "101110";   -- 6-bit two's complement
                                                         -- weight
  BEGIN

  c31 <= c31adder(weight, feature);     --call  user's function library to generate the first
                                            -- carry from column 3.
  c32 <= c32adder(weight, feature);  --  call user's function library to generate the second
                                            --carry  from column 3.
  c33 <= c33adder(weight, feature);  -- call user's function library to generate the third carry
                                            --from  column 3.
   s41 <= (weight(4) AND feature(0)) XOR (weight(3) AND feature(1));--the first row addition
   s42 <= s41 XOR (weight(2) AND feature(2)) XOR c31;  -- the second row addition
   s43 <= (weight(1) AND feature(3)) XOR s42 XOR c32;  -- the third row addition

        pl_4 <= s43 XOR c33;  -- the final result

  END  multiplication;
```

In this implementation, the partial product function block consists of 4 input bits and one output bit. The weight value is defined as a constant binary vector of 6-bit in two's complement format. The 4 input bits multiplied by the weight constant generates one bit of partial product output, which can be implemented in a half CLB function block. The advantage of this method is that it allows the weight value to be changed by entering the new weight statement and automatically synthesised into gate level logic using Viewlogic. Thus, updating a new weight value for a later modification can be an easy task without going deep into the actual FPGA logic design. Figure 6. 2 shows the automatically synthesised combination logic schematic of the weight multiplication for product bit number 4. The synthesised logic is packed into a CLB function generator. When all the partial product modules have been created by the VHDL synthesis tool, the outputs of the modules are added together to form the final product (Figure 5.2 in chapter 5).

The ripple carry adder design relies on the Xilinx soft macros library to control the carry propagation direction. These soft macros allow the designer to access carry logic easily and to control mapping and block placement. Figure 6.3 shows a 10-bit ripple carry adder design with fast carry logic implemented by using relationally placed macros (RPMs). This adder is used to add two partial products together. Each adder is designed with a carry logic mode macro and a function generator. As shown in Figure 6.3, the carry logic mode is used to generate the fast carry and the function generator to combine the carry with operands to form the sum. By using the Xilinx RPM, the ripple carry outputs are routed between CLBs on high-speed dedicated paths. A FORCE-0 (CY4_37) RPM is used to initialise the carry path to zero. As this FORCE-0 RPM only involves the carry logic, all the non-carry resources of the CLB are available for other uses. As shown in the schematic diagram, each CLB can implement two bit addition. Thus, 5 CLBs are required to implement the 10-bit ripple carry adder. The flip-flops in the CLB are inserted into the sum path to provide the pipelined design as described in the previous chapter. The summation for the partial product will not generate the overflow. Thus, no overflow prevention circuits are implemented at the significant bit. The outputs are fed into a clocked register for pipelining. In order to achieve high speed synchronous operation, the system clock is derived from a fast global driver. As each partial product bit needs only half a CLB, one weight multiplication requires 15 CLBs. It was estimated that, for the hidden layer, each node can be implemented in one XC4013, and for the output layer, the three nodes can be integrated into one FPGA device.

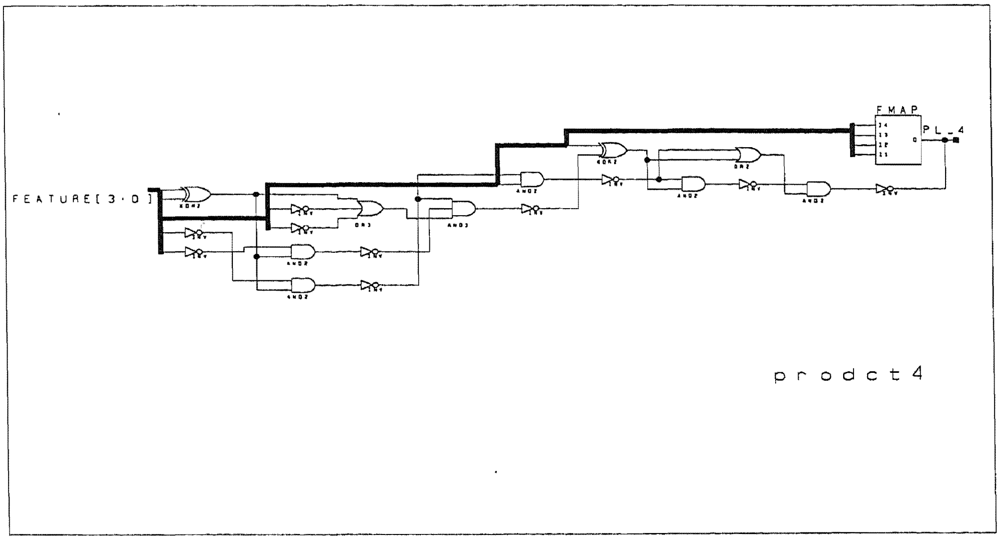


Figure 6.2: The partial product bit 4 schematic produced by VHDL synthesis

The plastic quad flat package (PQFP) XC4013PQ240-5 with speed grade of 5 is adopted for hidden neuron implementation. This is the highest density chip that the available development tool can support. A prior implementation results showed that each hidden node design uses 78% of the 575 CLBs and 92% of the 192 IOBs available. However, the design was failed during the PPR (Partition, Place and Route) process. The PPR tool generate a error message which shows that it cannot place the design. As the available tools could not support the placement and routed for complex design at the time, an effort to cut down the design to 54% CLB usage by using two FPGAs proved to be routable using available automatic place and routing tools. This suggested that the available placement and routing strategies in the software could not cope satisfactorily with the large arrays. Therefore, inefficiencies in the placement performed by the PPR tool prevent the necessary utilisation of the XC4013 resources. It may be possible to squeeze all of the design into a single chip by manual placement and routing. In this way, one could also minimise wire length to improve the performance, by forcing less critical nets onto the slower paths and using fast paths for the critical nets. However, the drawback of this optimisation method is not only that it is time consuming but also it causes design errors. With today's high density circuits and advanced CAD tools, this method appears to have less advantage. Moreover, it would exceed the time constraint on this project. It is expected that on going development tools will provide solutions for these problems (e. g. XACT version 6.1). Thus, instead of implementing one hidden node on a single FPGA, two chips are used to implement the design. One is XC4013PQ240 chip which has 240 pins and the other is XC4006PQ160 which has 160 pins. For the output layer design, three nodes are successfully implemented in a single chip of XC4013PQ240 with a 56% usage of the CLBs.

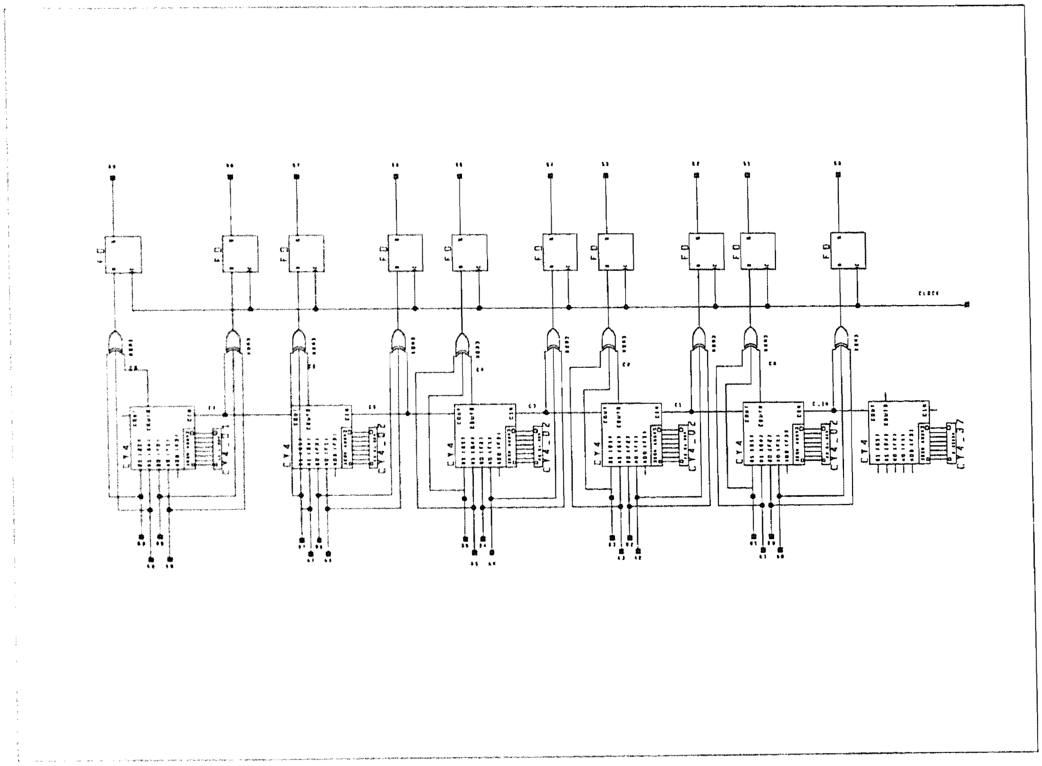


Figure 6.3: The 10-bit ripple carry adder with pipelined registers

6.3 Results

The XACT static timing analysis was used to estimate the "worst case" of time delay. The timing data from these paths gave a good indicator of the performance of both the technology and the circuit library. The maximum ripple carry adder in the adder tree is 18-bit. The carry propagates through 18 half CLB function blocks. Therefore, each function block has a time penalty associated with it. Figure 6.4 shows an Xdelay time analysis report. The delay of an 18-bit ripple carry adder was started from the initial carry-in CLB. The propagation delay for each function block is approximately 1.5 ns (nano-second). An 18-bit ripple carry adder with pipelined register has only 18.1 ns delay. This result indicates that the new

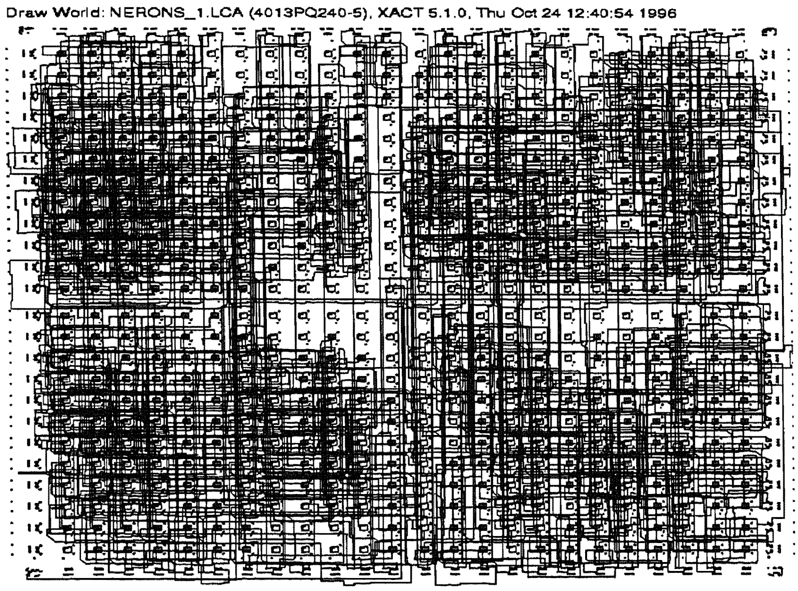
XC4000 architecture can provide sufficient performance for most time critical applications without special dedicated arithmetic design.

The design was fully implemented on the FGPAs and analysed with the static timing analysis tool. The worst time delay for the first hidden neural chip is 36.2 ns and for the second hidden neural chip is 36.3 ns. The maximum delay for the output neural chip is 33.9 ns. Thus, the maximum clock speed for hidden neurons is 27.6 MHz. The mapping result is summarised in Table 6.1. Comparing the final neuron routing result and 18-bit RCA timing result, one can see that there are delay factors that affect the design performance such as routing delay and I/O delay. As the nets, especially the long nets, need to go through a number of "configured switch boxes", that causes time delay. A careful route could avoid the longer delays by using long-lines rather than configured switch boxes. The layouts for the one hidden node and three output nodes implementations are shown in Figure 6.5 and Figure 6.6 respectively.

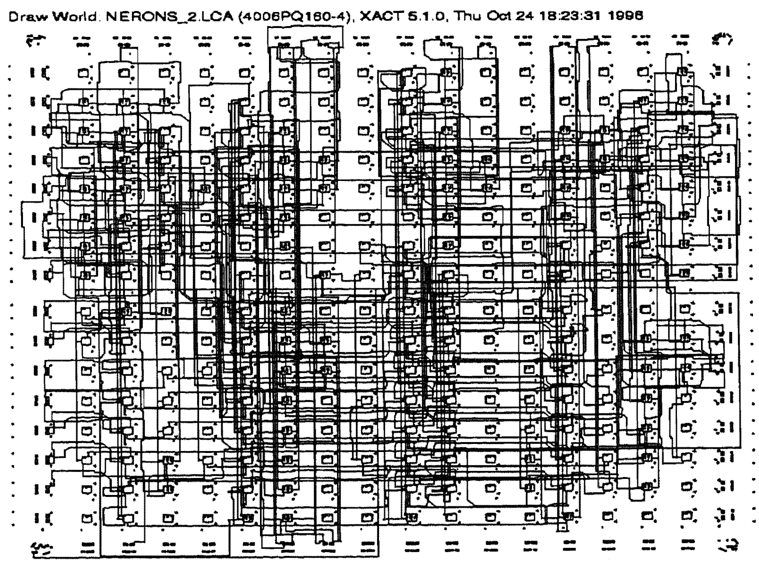
```
XDelay Report File: (4006PQ160-5), XACT 5.1.0,

From: Blk CLB1          to CLB_R24C7.COUT:   0.0ns ( 0.0ns)
Thru: Net $1I1/C_IN     to CLB_R23C7.CIN:    0.0ns ( 0.0ns)
Thru: Blk SD0           to CLB_R23C7.COUT:   1.5ns ( 1.5ns)
Thru: Net $1I1/C1       to CLB_R22C7.CIN:    0.0ns ( 1.5ns)
Thru: Blk SD2           to CLB_R22C7.COUT:   1.5ns ( 3.0ns)
Thru: Net $1I1/CB3      to CLB_R21C7.CIN:    0.0ns ( 3.0ns)
Thru: Blk SD4           to CLB_R21C7.COUT:   1.5ns ( 4.5ns)
Thru: Net $1I1/CB5      to CLB_R20C7.CIN:    0.0ns ( 4.5ns)
Thru: Blk SD6           to CLB_R20C7.COUT:   1.5ns ( 6.0ns)
Thru: Net $1I1/C7       to CLB_R19C7.CIN:    0.1ns ( 6.1ns)
Thru: Blk SD8           to CLB_R19C7.COUT:   1.5ns ( 7.6ns)
Thru: Net $1I1/C9       to CLB_R18C7.CIN:    0.0ns ( 7.6ns)
Thru: Blk SD10          to CLB_R18C7.COUT:   1.5ns ( 9.1ns)
Thru: Net $1I1/C11      to CLB_R17C7.CIN:    0.0ns ( 9.1ns)
Thru: Blk SD12          to CLB_R17C7.COUT:   1.5ns ( 10.6ns)
Thru: Net $1I1/C13      to CLB_R16C7.CIN:    0.0ns ( 10.6ns)
Thru: Blk SD14          to CLB_R16C7.COUT:   1.5ns ( 12.1ns)
Thru: Net $1I1/C15      to CLB_R15C7.CIN:    0.0ns ( 12.1ns)
    To: FF Setup (D), Blk SD16:  6.0ns ( 18.1ns)
Target FFX drives output net "SD16"
Target FFY drives output net "SD17"
Dest clock net : "CLK" (Rising edge)
```

Figure 6.4: A 18-bit ripple adder time delay analyse results

This individual neurons are ready to be download into physical components now. The proposed BacNet (Ballistocardiogram classification Network) can be integrated on a standalone board with interface components for BCG classification in real time. By making use of the advantage of the hardware reconfiguration capability, the proposed BacNet can be adopted for other medical field applications, such as biomedical image processing, by simply modifying the network weight parameters. The characteristics of the proposed BacNet are summarised in table 6.2



(A)



(B)

Figure 6.5: The layout of one hidden neuron in two FPGA chips. (A) is XC4013 and (B) is XC4006.

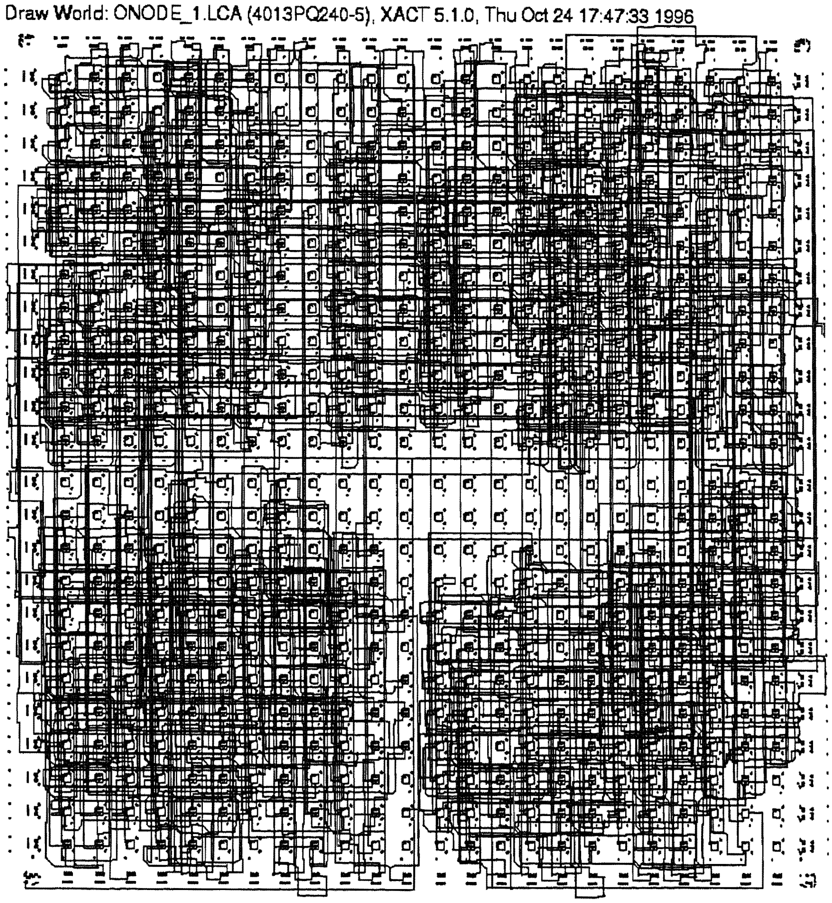


Figure 6.6: The layout of three output nodes in a single FPGA.

Table 6.1: Results of the Neurons Mapping

	One Hidden Node	One Output Node
Number of CLBs Used	440	94
Number of pins Used	171	34
FPGA Implementation	XC4013 & XC4006	1/3 of XC4013
Sigmoid Function	1K*8-bit PROM	None
Maximum path delay	36.3 nano-second	33.9 nano-second

Table 6.2: BacNet System Characteristics

Network Architecture	20 inputs, 4 hidden nodes and 3 outputs (20:4:3)
Number of Weights	99
Processing Parallelism	Fully Parallel
Maximum Clock Speed	27.6 MHz
Classification Speed	27 MPPS (million pattern per second)
Computation Speed	2.7 BCPS (billion connection per second)
Input Features	2's complement; -127 to +127
Weights	2's complement; -32 to +32
Hidden Layer Node Outputs	2's complement; -127 to +127
Output Layer Node	Logic 1 and 0
FPGA Chips Required	5 XC4013 and 4 XC4006

6.4 Discussion

6.4.1 Implementation

Inefficiencies in the placement performed by the available development tool prevents the necessary utilisation of the XC4013 resources and so the placement of the single hidden neuron had to be implemented in two FPGA chips. By reviewing the die layout of the hidden neuron chip in Figure 6.5, one finds that the automatic tool placed most of the design on the four corners and there are still routing resources available in the middle of the area. This suggests that because middle area function blocks need long wires to connect to I/O which could consume more routing and decrease the performance, the trade-off of the automatic place and routing tool leaves the middle area unrouted. This result indicates that there is a possibility that a single hidden neuron can be routed in one XC4013 chip by carefully routine the design in the middle area of the chip. The improvement in placement density to fulfil the potential of the XC4000 series could be achieved by using the new updated place and routing tool (e.g. XACT 6.1) or another development tool (e.g. IBM BooleDozer). These modern development tools support floorplanning design features which allow the user to iteratively improve design compactness and performance.

For a fixed size network, the available parallelism is set by the number of inputs and neurons. Theoretically, the techniques described in this chapter and chapter 5 can be expanded to a large size neural network implementation by using higher density FPGAs and a high degree parallelism to make it attractive in speed. In building such a system, one must observe the constraints imposed by the I/O and the number of chips. The input bandwidth required of each FPGA in such a system might be too high, even though it has adequate internal logic density. On the other hand, it should be noted that, in SRAM based FPGA design, one has to trade usable logic gates for routing resources. These resources can constrain the design routed on several chips. Thus, a large parallelism could increase the circuit board size which makes it physically bulky. For a large size neural network implementation, a balance between the necessary requirement and the cost can be achieved using serial computation technology. The advantage of the bit-serial design is that it requires less area than parallel implementations. For example, Eldredge and Hutchings (1994) show that one neuron can be implemented on a single XC3090 FPGA with bit-serial implementation. Design experience suggests that one should take both resources and performance into consideration for an application.

6.4.2 Performance

Hardware parallelism with a performance of 27 MPPS under a 27 MHz system clock was realised through automatic placement and routing. The experiments demonstrated that the built-in special carry logic of the XC4000 device provides high speed for arithmetic units such as a ripple carry adder. The delay of a RCA mainly depends on the number of adder bits. It is suggested that long line routing is one of the main factors affecting the system performance.

Table 6.3 provides the performances of some implementations using custom technology, FPGAs and DSP technology reported in the literature. It is difficult to directly compare these performances, because of the variety of characteristics involved in the implementation technology and the degree of parallelism, for example, calculation precision range, neuron parallelism, input parallelism, or bit parallelism. Some typical implementations reported in the literature are selected in the Table. Furthermore, the speed measurement of connections per second (CPS) is not a good measure of parallel machine performance, since this criterion is so dependent on the size of the network (number of weights). Thus, the number of patterns per second (PPS) is used regardless of pattern size or network size.

Table 6.3: Some Neural Network Implementations

System	Number of Neurons	Peak Performance	Weight Precision	Implementation Technology
Digital Chip (Watanabe et al., 1993)	1024	500 KPPS	8 bits	Single CMOS chip
ANNA (Boser et al., 1992)	16-256	20 MPPS	5 bits	Single CMOS Chip
GANGLION (Cox & Blanz, 1992)	30	20 MPPS	8 bits	36 FPGAs chips
FPGA Network (Botros & Abdul-Aziz, 1993)	11	70 KPPS	8-bit	10 FPGAs chips
MUSIC (Muller et al., 1995)	1530	30.4 KPPS	32-bit floating point	40 DSP chips

The digital Chip based on a DRAM memory is design by Watanabe et al (1993). The chip was fabricated with 0.5-μm CMOS technology with 1.5 -V operation. An one-transistor DRAM cell array is used to store synapse weights to reduce the chip area. The arithmetic precision is 8-bit. The chip contains 256 parallel processing circuits and can be used for up to 1024 neuron implementation. The low power consumption made this chip suitable for portable instrument application.

Artificial neural network ALU (ANNA) is proposed by Boser et al. (1992). This special design is customised to a precision reduction application in which the input is 3-bit and the weight is 5-bit. Although processing element parallelism is realised on the chip, the input is sequential to reduce the chip input I/O. The chip can be configured as 16 neurons with 256 inputs or 256 neurons with 16 inputs. Due to the reduction in computation precision, the chip has a high performance.

The Multiprocessor System with Intelligent Communication (MUSIC) aims to provide a high performance and low cost system for neural model simulation (Muller et al., 1995). The parallel processing elements of 63 Motorola DSP 96002 chips are implemented on a 21 board system. Each board contains 3 DSP and a board interface manager (Transputer T805). All the weights are stored in fast RAM. The floating point precision used is 32-bit. The system can perform continuous weight updates with backpropagation learning.

The FPGA based parallel neural network implementation such as GANGLION (Cox and Blanz, 1992) and FPGA Network (Botros and Abdul-Aziz, 1993) are closely related to the work reported in this chapter. Both GANGLION and FPGA network used the same XC3000 series FPGA. The different implementation methods achieved different performances. Although the FPGA Network was implemented in a parallel processing fashion, the sequential accumulator design degraded the system performance. On the other hand, GANGLION was implemented with a dedicated parallel adder tree to improve the system performance. Compared with the XC3000 series design, the proposed BacNet implementation based on XC4000 series can achieve a higher performance at 27 MPPS using a simple ripple carry adder tree. Furthermore, the design with ripple carry adders is simple and the VHDL method eases the weight parameters update. Further performance improvement can be achieved by using dedicated arithmetic unit design.

6.4.3 Flexibility

As the weight multiplication was realised by using VHDL statement, the weight parameters change is quite straightforward by simply change the binary value. The synthesis time involved is PC dependent. On a 486-33 PC, a couple of hours is required to synthesis the total 46 weight multipliers. These weight multiplication modules can be placement controlled to replace the old weight multiplication hardware. Thus, a new system can be prototyped within a couple of hours to update to new weight values when the network can be retrained with additional data set. Furthermore, the ripple carry adder allows easy expansion in the schematic design. For new applications that requires different neural architectures, the hidden and output neurons can be easily expanded with same design approaches to adapt to new requirements.

6.5 Conclusions

In this chapter, a method for mapping a feedforward neural network to Xilinx FPGAs is presented. The design is based on the combined VHDL and schematic method, thereby, future weight updating procedure is simplified. Hardware parallelism is achieved using an automatic place and routing tool. Due to the inefficiency of available development tools and resources, one hidden neuron is implemented in two devices. However, it is shown that it is feasible to realise a one neuron per chip solution on the XC4000 series. The simulation results show that the implemented neurons are capable of operating at a 27 MHz system clock and provide 27 million processes per second performance. This result supports the belief that a compact and improved performance network implementation can be achieved by making use of improved FPGA architecture. By comparing to the previous design based on XC3000 series, it is shown that using improved FPGA architecture for neural network implementation simplified the hardware design and provided a superior performance. It is also demonstrated that parallel hardware based on FPGAs can deliver a high performance and flexibility for real time applications. The design method described in this study can also be extended to other biomedical signal and image processing fields. It is indicated that the development tools have an impact on optimal design and high performance for SRAM based devices.

Chapter 7

Conclusions and Suggestions for Future Work

7.1 Conclusions

7.1.1 Summary

The work presented in this thesis was originally motivated by the desire to investigate the capabilities of artificial neural networks for automatic BCG classification in portable medical instruments. The primary reason for this interest is the feature that no electrodes need to be attached to the body during measurement. It provides a potential application, in that the BCG can be used as a relatively low cost, non-invasive, easy-to-use, home screening procedure to find pathological signs as early as possible and to treat at the earliest possible stage. However, previously reported work for BCG analysis relies on an off-line two stage analysis method; recording the BCG signal in the clinical environment and playing records back to digital computers for analysis. . This off-line method is suitable for laboratory study, but not suitable for home care applications. The time and labour involved from measurements to analysis are expensive and inconvenient for the patients. Furthermore, most of the published works are based on traditional statistical analysis and heuristic methods. Therefore, the issues of using improved signal processing techniques, such as state-of-art pattern recognition techniques and neural computing, need to be better addressed in the BCG signal analysis field. There is only one reported work using artificial neural network for BCG signal analysis which was conducted by Zhao (1994). He demonstrated a framework of a neural based classification system using the combination of a statistical method and a feedforward MLP technique for BCG classification. However, the feature extraction is based on the whole data set and the classifier is trained with the leave-one-out method. The generalisation ability of the system to unseen data is in doubt. This makes the method only suitable for analysis of collected BCG data in the Laboratory so that it could not be applied for real time applications.

The presented work in this thesis can be considered as the first step toward developing an on-line BCG classification system for portable instrument applications. The main contributions of this work can be summarised as: (1) two new classification models for real time automatic classification of time domain BCG signal are developed and evaluated; (2) a methodology for implementing the two stage classification systems for automatic BCG signal analysis is presented; (3) a new hybrid learning algorithm is developed for weight precision selection of the feedforward multilayer perceptrons to reduce the hardware implementation complexity; (4) an approach of off-line training and on-line implementation of a neural classifier using FPGAs for BCG signal processing instrument is presented. The framework presented in this thesis also provides a further motivation for the study of neural networks based systems for automatic BCG signal analysis and instrument development. It is believed that the methods presented can be easily generalised to other biomedical signal processing fields.

7.1.1.1 Review of the Work

This presented work can be summarised as follows:

1. A neural network for principal component analysis is introduced and applied to BCG feature extraction to overcome the problem that a limited data set and a relatively small number of heart attack subjects in the data samples can degrade generalisation performance of the neural classifier. This method allows the features to be generated automatically from training data using an unsupervised learning rule network.

2. A combination of a PCA network and a feedforward MLP system is developed for BCG classification. Both linear PCA and non-linear PCA networks have been studied. 5 PCA network outputs and 10 PCA network outputs are used for the neural classifier in the experiments. The performance of the neural classifier is evaluated by leave-one-out and data partitioning schemes. The best classification on heart attack subjects was achieved with a 10 output non-linear PCA network with β=2.

3. A methodology for off-line training and on-line implementation is established for the combined neural networks classification system implementation. This method implements the neural models with less complex design for portable instrument applications.

4. A multiresolution wavelet technique is introduced for the first time in BCG signal analysis. By choosing appropriate resolution signals to be input to the neural classifier, a dimensionality reduction is achieved. This method requires no prior knowledge of the statistical distribution of data samples (c.f. principal component analysis, Kittler-Young transform).

5. A system combining multiresolution wavelet transform and MLP classification is developed. The experimental results indicated that using 20 coarse components as neural classifier inputs, the combined multiresolution wavelet transform and MLP system has much better performance than the combined PCA network and MLP scheme.

6. A method for realising a rapid classification instrument for real time applications is presented. The proposed system mainly consists of three function modules; signal pre-processing, wavelet transform and neural classifier.

7. A hybrid learning algorithm is developed to reduce the weight precision of the neural classifier. This allows the neural network to be implemented on limited precision hardware to reduce the hardware design complexity. Two benchmark experiments demonstrate the effectiveness of the hybrid learning algorithm in convergence ability for low weight precision learning compared to the standard backpropagation rule.

8. A flexible hardware platform using field programmable gate arrays is introduced into the BCG instrument development field. A method of implementing a fully parallel neural network on Xilinx's FPGAs (XC4000) is described. The work demonstrates that reconfigurable hardware allows hardwired parameters to be updated as further data is gathered, which is a common case in medical applications. It also provides a low-cost and fast design cycle solution in the biomedical application field using semi-custom circuit design to achieve high performance.

7.1.2 On-Line Approaches for BCG Classification

7.1.2.1 A Combined PCA Network and MLP Classification Model

Due to the limited number of BCG samples in this study, the reduction of the number of input features to the neural network to improve neural classifier performance forms the main consideration in developing a classification system for practical applications. In a previous work, Zhao (1994) has demonstrated that a statistical method, K-Y transformation, has the capability of transforming the 80 dimension BCG input into a 3 dimensional feature input to the neural classifier. This classification scheme achieved good classification results on heart attack subjects. As discussed in chapter 1, the feature extraction is based on the statistics of the whole data set so that the result is not optimal in a theoretical sense (Lee and Landgrebe, 1993). Furthermore, the neural classifier is evaluated with a leave-one-out method so that it could not generalise to new cases in real applications. However, the capability that the statistical approach showed in feature reduction was encouraging. Along this line, a computationally efficient classification model based on the combination of a PCA network and a MLP is proposed for automatic BCG classification.

As discussed in chapter 1 and chapter 2, the conventional PCA or the related Karhunen-Loeve transform are performed by calculating the covariance matrix and analysing its eigenvectors which involves substantial computation time on a sequential computer. Moreover, the approach is not suitable for the situation where data come incrementally or in the on-line way (Xu and Yuille, 1995). The PCA network using an unsupervised learning rule, however, can overcome these limitations. The PCA network has the ability to extract the principal components directly from the data and it provides the capability to process the incoming data in real time. The parallel property of the network enables the hardware implementation to speed up the performance. A method for implementing the combined neural network system for portable instruments is presented. This method is based on the off-line training and on-line implementation strategy in order to reduce the design complexity and system cost in a portable instrument.

The BCG signal pre-processing technique proposed by the Medical Informatics Unit of the University of Cambridge is adopted throughout the study. It consists of filtering noise, signal averaging and normalising the time domain signal to a standardised length. The performance of the proposed PCA network and MLP model was compared with the scheme using the K-Y transform and MLP (Zhao, 1994). The experiments show that the statistical method for feature extraction was not satisfactory with the data set under investigation. The important effect on the generalisation performance of such a combined system is that a limited data set and unbalanced classes in the training set can cause a bias in the statistical estimation during the feature extraction phase (Hoffbeck and Landgrebe, 1996). This raises a difficult problem for the validation of the proposed system's ability to generalise. The results of this study suggest that the statistical method for feature extraction could not be the optimal approach to the data set under study. The results indicate that a careful attention should be paid when using statistical methods for feature extraction with a limited data set and unbalanced classes in the training set.

Although experimental results show that the proposed PCA networks and MLP systems have not achieved the expected performance, one of the most important results is that it demonstrates a practical way for implementing a combined neural network classification model for real time applications. Furthermore, it has demonstrated an approach for BCG signal analysis using PCA neural networks which has always been done with conventional PCA methods in the literature. The presented PCA network method leads to suggest a practical way for adaptive BCG signal processing, for example, BCG signal compressing and clustering. By implementing the PCA network in hardware (Kotilainen, et al., 1992), it is feasible to use a PCA network to process high volume data such as vectorballistocardiography in real time. Moreover, the work using non-linear PCA networks presented in chapter 2 also provides a potential application area that analyses signals with high order statistical methods (Karhunen and Joutsensalo, 1995). Overall, the classification model proposed in this study offers the interest for future investigation when the database is expanded.

7.1.2.2 A Combined Wavelet Transform and MLP Classification Model

The simulation experiments indicate that the limited data set is a severe limitation in this study. Therefore, multiresolution wavelet transformation which has not previously been used for BCG feature dimension reduction, is explored. It is based on the orthonormal wavelet theory that an arbitrary signal can be represented exactly as a weighted sum of wavelet basis functions. By using the wavelet transform, one can project the raw BCG signal into a low dimension space and decompose the original signal into a series of sub-signals at different scales. If the amplitudes and timing of different peaks of a BCG signal can be preserved at a low resolution, a BCG feature reduction can be achieved. One of the advantages of the wavelet transform is that it requires no prior knowledge of the statistical distribution of data samples. Therefore, this allows further focus on the improvement of MLP generalisation. Furthermore, due to pioneer research work by Daubechies (1990) and Mallat (1989), the wavelet transform can be easily implemented as a FIR filter.

A comprehensive performance evaluation of the proposed system is presented using the available data set. Compared to the combined PCA network and MLP classification models, the combined wavelet transform and MLP system provides much better performance than the combined PCA network and MLP model and has a comparable performance to the combined K-Y transform and MLP scheme. In regard to performance, memory and computation cost, the combined wavelet transform and MLP model has clear advantages over those BCG classification schemes considered. The comparison results indicates that effectiveness and performance of wavelet transform method is better than the simple signal resampling scheme which was tested by Zhao (1994). To date, the choice of the most appropriate wavelet function is still an open research issue. The orthonormal wavelet developed by Mallat (1989) has proved quite successful in this application.

In the biomedical signal processing field, there have been few attempts to apply wavelet transforms and neural networks to solve signal classification problems. Most of the work reported in the literature uses wavelet transforms for signal compression and noise filtering. This study is the first to attack the BCG classification problem with a combination of wavelet transform and neural network approaches, and is the first research to apply wavelet transform method to the BCG signal analysis field. This research suggests that the wavelet transform offers an attractive alternative method in BCG signal analysis. The wavelet transform provides an efficient way to reduce the signal resolution of interest for analysis. Therefore, it offers a practical approach to solve real world problems. The work presented in this study can be considered as a motivation for future study to compare the classical signal processing method such as fast Fourier transform to this signal processing technique on BCG signal analysis.

Two common methods for improving network generalisation are discussed and evaluated in chapter 3; one is to use a weight decay technique and the other is to add noise to the training data set before training the network. The evidence suggests that the performance improvement is closely related to the training samples. If the data set is small, these techniques are not applicable. Overall, a methodology for implementing a real time BCG classification system for portable instrumentation is presented in chapter 3. The system mainly consists of three functional modules; signal pre-processing, wavelet transform and neural classifier. The proposed system has the advantage of being computationally efficient for real time applications. It is believed that the methods presented can be easily generalised to other BCG classification applications, such as on-line sleep classification study for the BioMatt Monitoring System (BIOREC, 1994).

7.1.3 Hardware Implementation of Neural Networks

Neural networks have increasing numbers of applications. But hardware implementation is still not as popular as the use of software tools for neural network applications. As more and more practical biomedical signals and image processing require more computational resources to process the demanding volume of data, speed is often the most critical factor determining the usefulness of a solution. These problems can be solved by taking full advantage of parallel processing. On the other hand, in many portable medical instrument applications, it is required that the system should be able to indicate the class of a vector, representing both input and output as an electrical signal in a short time. It is aimed to demonstrate an approach to implement a neural network on flexible hardware which can be extended to other biomedical signal and image processing fields where a high speed is required. The presented work is to explore a fully parallel neural network design with FPGAs using an off-line training and on-line implementation approach.

7.1.3.1 Weight Precision Selection with a Hybrid Learning Algorithm

One of the issues in the design of neural network hardware is the precision of the synaptic weights. A high precision weight requires a big multiplier and a large weights storage which occupy large areas on digital chips. On the other hand, a high speed can be achieved with limited precision in weights. The standard backpropagation algorithm has the disadvantage for limited weight precision learning that when the weight precision becomes low, the updating weight change could vanish and the network could be stuck in a paralysis situation so that learning could not go further.

A hybrid learning algorithm for low weight precision network training is developed to overcome this difficulty in chapter 4. The algorithm is different from reported work in that it makes use of the advantages of two algorithms to overcome the network paralysis problem and to improve the learning process. The random optimisation algorithm has the capability of global search even in low weight precision learning, but is rather slow to reach the solution. Standard backpropagation has a good capability for fine tuning the solution. By effectively using the random optimisation algorithm, an efficient weight update mechanism can be achieved. The simulation results on two benchmark problems are used to compare the effectiveness of the hybrid algorithms and the standard backpropagation rule. The experiments indicate that the new hybrid learning algorithm is an improvement on the standard backpropagation algorithm both in convergence capability and robustness. Furthermore, a training strategy (Yu, et al., 1996) for an extension of the work of Nakayama et al. (1990) for low precision learning with small data sets, and the hybrid learning algorithm is applied to the BCG neural classifier weights precision selection. The results demonstrate that the generalisation capability of a MLP with 6-bit weight precision provides comparable performance to the network with floating point computations.

One advantage of the proposed learning rule is that it is easy to implement and the additional calculations involved for the modification are relatively small. The second is that it has high speed of convergence and robustness compared with the conventional backpropagation rule. These features offer the potential application that it is possible to further modify the proposed algorithm for on-line learning with limited precision. Finally, the introduced random optimisation method enables the network to escape local minima during the network training. The experimental results suggest that the developed training strategy is able to cope with the small data set problem. This algorithm can generally apply to those standard feedforward neural networks for weight precision reduction. In this study, the constant learning rate is adopted. However, the adaptive learning rate scheme could be considered in future to further improve the convergence speed. Furthermore, a comprehensive comparison of the proposed hybrid algorithm with other modified learning schemes (e.g. Dundar and Rose, 1995; Alibeik et al., 1995) would merit future research.

7.1.3.2 Hardware Implementation

As discussed in chapter 5, in many real world problems, a specific hardware parallelism is required to deal with high volume signal processing in real time. Flexibility, cost and fast design cycle are also major factors for hardware implementation, especially for instrument prototype stage. Taking these design issues into consideration, it suggests that FPGAs are an ideal platform for this study. In fact, it is not realistic to obtain an adequate data set in a limited time in most biomedical instrument studies. Therefore, a flexible design is essential for these applications that require the parameters to be modified later to improve instrument performance. To this end, a case study is presented for placing hidden and output neurons on FPGAs (chapter 5; chapter 6).

The work can be considered as an extension to the work of Cox and Blanz (1992), and Botros and Abdul-Aziz (1994). In this study, an improved SRAM based architecture FPGA, and combined VHDL and schematic design are used to implement neurons. Due to the inefficient FPGA implementation tools, one hidden node is mapped into two FPGA chips. For the output layer, three nodes are successively mapped into one XC4013 chip. Five XC4013 and four XC4006 FPGA chips are totally required for implementing a 20 inputs, 4 hidden nodes and 3 outputs network with 8-bit inputs and 6-bit weights. . The analysis results suggest that it would be possible to achieve a one neuron per chip solution on XC4013 with improved placement and routing tools. Thus, the network size could be further reduced to five XC4013 chips. Although resources were not available to transfer the design into hardware, the neurons implementation and time analysis results suggest that the proposed design has a maximum 37 nano-second delay which means that it is capable to generate 27 million pattern per second performance. This performance is much better than the previous reported work (Cox and Blanz 1992; Botros and Abdul-Aziz 1994).

The weight multiplication design with VHDL technology shows that the weight parameters can be easily changed without deeply involved hardware knowledge. The synthesis time involved for updating new weight parameters takes only a couple of hours. As the VHDL based design offers clear advantages over traditional schematic design methods, this work could be considered as a basis for future HDL-based design of neural networks and wavelet transforms. In fact, the demonstrated design method is intended to be sufficiently general that it can be easily applied to a wide variety biomedical signal and image processing applications. Furthermore, it demonstrates a method of implementing the neural classifier in low precision hardware for real world applications in the BCG signal processing field. A comparison with some specifically designed neural chips shows that FPGA based neural implementation offers a competitive performance. Overall, this work demonstrates a flexible hardware design approach dealing with the problems in biomedical instrument developments where the database expands with time. The design method and the proposed system performance presented here leave room for further development and improvement.

7.2 Suggestions for Future Work

It has been noted throughout this study that the small number of BCG data samples, especially for the few heart attack patterns in the data samples, causes an estimation problem in the generalisation capability of the classification models. To fully evaluate the prognostic value of the proposed classification system, a large scale trial using the proposed classification system must be conducted. However, a sufficient fund is required to cover the time and resources cost to achieve this goal. Furthermore, it would be interesting to include more parameter measurements such as body temperature and other cardiac conditions, for example, heart rate variability, into the proposed system for home cardiac function assessment and monitoring applications. A further research of these parameters associated with the prognostic values could make the instrument more useful.

Wavelet transformation provides an interesting subject for future research in BCG features reduction. A further subject for future research is the experimental investigation of selecting appropriate wavelet basis functions to improve signal approximation accuracy. A comparison of wavelet transform with other conventional signal processing methods would also merit future research interest. Moreover, applying wavelet transform for BCG signal normalisation and comparison with other approaches (e.g. multirate signal processing) forms an interest direction for future investigation.

In medical applications, the subsequent identification and elimination of patients with false positives is an unacceptable factor (Ream, 1978). Therefore, research is needed into ways of increasing required classification performance to accommodate the misclassification rate. To further improve the MLP classifier performance, investigating improved learning strategies with small training data sets will form one of the subjects for future research to obtain a better performance. Along this line, using an improved learning rule such as the learning algorithm proposed by Jeong and Lee (1996) provides a important subject for future research. Furthermore, another direction for future investigation is to improve classification reliability by taking a decision to reject an uncertain sample instead of running the risk of misclassifying it (Cordella et al., 1995). This can keep misclassification rates as low as possible by increasing the reject rate. The technique of improving data pre-processing for the classifier training also provides opportunities for further research, such as selecting concise training sets from data samples (Plutowski and White, 1993).

A further extension to future research arising from chapter 4 is the experimental comparison of the proposed hybrid learning algorithm with the alternative available approaches suggested in section 4.6. Along with the experimental evaluation of the hybrid learning algorithm, it would be interesting to further modify the algorithm for on-line training implementation in VLSI. This would improve the usefulness of the instrument for real time applications.

The advance in FPGA technology has always posed a design challenge in trades-off between cost and speed. A further subject for future research is the consideration of the development of a flexible portable BCG classification system, with in-system dynamic reconfiguration capability for on-line neural network training and BCG data compression algorithms implementation to improve the instrument efficiency and capability in real time. An alternative approach to future research is to use HDL based design to implement the classification model to make the system expansion and modification more convenient. Initial work has been explored by the author using Altera's SRAM based device (Neale, 1996).

The trade-off of the proposed portable system is that the raw data would not be available for further analysis or for use with more powerful algorithms at the hospital. One solution would be to have in-system memory to store the raw data and send this data to the hospital later via a telephone line. Another direction is to research a transmission technology for the portable monitoring system to send the data to a central hospital computer. This can be particularly appropriate in "hostile" environments.

Bibliography

Alibeik, S.A., Nemati, F., and Sharif-Bakhtiar, M., "Analog feedforward neural networks with very low precision weights," in *Proceedings of 1995 IEEE International Conference of Neural Networks*, pp. 90-94, 1995

Alihanka, J., "*Basic Principle for Analyzing and Scoring Bio-Matt (SCSB) Recordings*," Typopress Oy, 1987

Alihanka, J., Vaahtoranta, K., and Saarikivi, I., " A new method for long term monitoring of the ballistocardiogram, heart rate, and respiration," *American Journal of Physiology*, Vol. 240, pp. 384-382, 1981

Alpsan, D., Towsey, M., Ozdamar, O., Tsoi, A.G., and Ghista, D.N., "Efficacy of modified backpropagation and optimization method on a real-world medical problem," *Neural Networks*, Vol. 8, No. 6, pp. 945-962, 1995

Altera, "Implementing FIR Filters in FLEX Devices", *Altera Application Note 73*, Altera Corporation, 1996

Anderson, J.A., "Two models for memory organization," *Mathematical Biosciences*, Vol. 8, pp. 137-160, 1970

Baba, N., "A new approach for finding the global minimum of error function of neural networks," *Neural Networks*, Vol. 2, pp. 367-373, 1989

Bade, S.L. and Hutchings, B.L., "FPGA-based stochastic neural networks-Implementation," in *Proceedings of IEEE Workshop on FPGAs for Custom Computing Machine*, pp. 189-198, 1994

Baldi, P.F. and Hornik, K., "Learning in linear neural networks: a survey", *IEEE Transactions on Neural Networks*, Vol. 6, No. 4, pp. 837-858, 1995

Bancroft, W.H., Tucker, M., Jackson, D.H., and Eddleman, E.E., "Automatic computer processing of ultralow frequency ballistocardiograms," *Bibl. Cardiol.*, Vol. 32, pp. 1-10, 1973

Bannour, S., and Azimi-Sadjadi, M.R., "Principal component extraction using recursive least squares learning," *IEEE Transactions on Neural Networks*, Vol. 6, No. 2, pp. 457-469, 1995

Barranco, B.L., Sinencio, E.S., Rodriguez, A., and Huertas, J.L., "A CMOS analog adaptive BAM with on-chip learning and weight refreshing," *IEEE Transactions on Neural Networks*, Vol. 4, No. 3, pp. 445-455, 1993

Battiti, R. and Masulli, F., "BFGS optimization for faster and automated supervised learning," in *Proceedings of the INNC'90*, Kluwer Acdemic publishes, 1990

Baugh, C.R. and Wooley, B.A., "A two's complement parallel array multiplication algorithm," *IEEE Transactions on Computers*, Vol. c-22, No. 12, pp. 1045-1047, 1973

Baum, E.B. and Haussler, D., "What size Net gives valid generalization," *Neural Computation*, Vol. 1, pp. 151-160, 1989

Becher, S. and Le Cun, Y., "Improving the convergence of backpropagation learning with second-order methods," in *Proceedings of Connectionist Models Summer School*, pp. 29-37, 1988

Bertolet, B.D. and Pepine, C.J., "Clinical development of a miniature digital ambulatory ECG Transient myocardial ischaemia detection system," *Journal of Electrocardiology*, Vol. 26, pp. 20-24, 1994

BIOREC, "*BioMatt Monitoring System*," BIOREC Co., 1994

Bortolan, A. and Willems, J.L., "Diagnostic ECG classification based on neural networks," *Journal of Electrocardiology*, Vol. 26, pp. 75-79, 1994

Boser, B.E., Sakinger, E., Bromley, J., Iecum, Y., and Jackel L.D., "Hardware requirements for neural network pattern classifiers," *IEEE Micro*, Vol. 12, No. 1, pp. 32-40, 1992

Botros, N.M. and Abdul-Aziz, M., "Hardware Implementation of an artificial neural network," in *Proceedings of IEEE International Conference on Neural Networks*, pp. 1252-1257, 1993

Botros, N.M. and Abdul-Aziz, M., "Hardware implementation of an artificial neural network using field programmable gate arrays (FPGA's)," *IEEE Transactions on Industrial Electronics*, Vol. 41, No. 6, pp. 665-667, 1994

Bout, D.E.V.D., Morris, J.N., Thomae, D., Labrozzi, S., Wingo, S., and Hallman, D., "AnyBoard: An FPGA-based reconfigurable system," *IEEE Design and Test*, Vol. 12, No. 1, pp. 21-30, 1992

Brouwer, R.K., "A method for training recurrent neural networks for classification by building basins of attraction," *Neural Networks*, Vol. 8, No. 4, pp. 597-603, 1995

Brown S. and Rose, J., "FPGA and CPLD architectures: A Tutorial," *IEEE Design and Test of Computers*, Vol. 13, No., 2, pp. 42-57, 1996

Brown, H., DeLalla, V., Epstein, M., and Hoffman, M., "*Clinical Ballistocardiography*," The Macmillan Company, New York, 1952

Brown, S., Francis, R., Rose, J., and Vranesic, Z., "*Field-Programmable Gate Arrays*," Kluwer Academic Publishers, USA, 1992

Buemi, M., Macerata, A., Taddei, A., Varanini, M., Emdin, M., and Marchesi, C., "Monitoring patients through principal components analysis," *in Proceedings of IEEE International Conference of Computer in Cardiology*, pp. 385-338, 1992

Chakrabarti, S., Bindal, N., and Theagharajan, K., "Robust radar target classifier using artificial neural networks," *IEEE Transactions on Neural Networks*, Vol. 6, No. 3, pp. 760-766, 1995

Chen, D.C. and Rabaey, J.M., "A reconfigurable multiprocessor IC for rapid prototyping of algorithmic-specific high-speed DSP data paths," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 12, pp. 1895-1904, 1992

Chen, J., Itoh, S., and Hashimoto, T., "ECG data compression by using wavelet transform," *IECE Trans. Inf. & Syst.*, Vol. E76-D, No. 12, pp. 1454-1461, 1993

Christensen, G.E., Miller, M.I., Vannier, M.W., and Grenander, U., "Individualizing neuro-anatomical atlases using a massively parallel computer," *Computer*, Vol. 29, No. 1, pp. 32-38, 1996

Cody, M.A., "The fast wavelet transform", *Dr. Dobb's Journal*, pp. 16-101, April, 1992

Cordella., L.P., Stefano, C.D., Tortorella, F., and Vento, M., "A method for improving classification reliability of multilayer perceptrons," *IEEE Transactions on Neural Networks*, Vol. 6, No. 5, pp. 1140-1147, 1995

Courrieu, P., "A convergent generator of neural networks," *Neural Networks*, Vol. 6, No. 6, pp. 855-844, 1993

Cox, C.E. and Blanz, W.E., "GANGLION-A fast field-programmable gate array implementation of a connectionist classifier," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 3, pp. 288-299, 1992

Crowe, J., Gibson, N., Woolfson, M., and Somekh, M., "Wavelet transform as a potential tool for ECG analysis and compression," *Journal of Biomedical Engineering*, Vol. 14, pp. 268-272, 1992

Cura, T., "Estrogen: Key Player in Heart Disease Among Women," *Science*, Vol. 269, pp. 771-773, 1995

Dargie, H.J, "Glasgow Gets to Grip with Heart Failure," *MRC News*, Vol. 4, pp. 12-15, 1994

Dassen, W.R.M., Mulleneers, R.G.A., Dulk, K.D.D., and Tamlmon, J.L., "Artificial neural networks and ECG interpretation," *Journal of Electrocardiology*, Vol. 26, pp. 61-65, 1994

Daubechies, I., "Orthonormal bases of compactly supported wavelets," *Communication on Pure and Applied Mathematics*, Vol. XLI, pp. 909-996, 1988

Deller, J.R and Hunt, S.D., "A simple 'linearized' learning algorithm which outperforms backpropagation," in *Proceedings of the International Joint Conference on Neural Networks '92*, Baltimore, 1992

Denoeux, T. and Lengelle, R., "Initializing back propagation network with prototype," *Neural Networks*, Vol. 6, pp. 351-363, 1993

Devijver, P.A. and Kittler, J., "*Pattern Recognition: A Statistical Approach*," Prentice-Hall, 1982

Dundar, G. and Rose, Kenneth, "The effects of quantization on multilayer neural networks," *IEEE Transactions on Neural Networks*, Vol. 6, No. 6, pp. 1446-1451, 1995

Eldrege, J.G. and Hutchings, B.L., "Density enhancement of neural network using FPGAs and run-time reconfiguration," in *Proceedings of IEEE Workshop on FPGAs for Custom Computing Machine*, pp. 180-188, 1994

Erkinjuntti, M., Vaahtoranta, K., Alihanka, J., and Kero, P., "Use of the SCSB method for monitoring of respiration, body movements and ballistocardiogram in infants," *Early Human Development*, Vol. 9, pp. 119-126, 1984

Farrow, D., "Using programmable logic cell arrays in a satellite earth station," *VLSI Systems Design*, Vol. 8, No. 4, pp. 26-32, 1987

Farrugia, S., Yee, H., and Nickolis, P., "Implantable cardioverter defibrillator Electrogram recognition with a multilayer perceptron," *PACE*, Vol. 16, No. 2, 1993

Finnoff, W., Hergert, F., and Zimmermann, H.G., "Improving model selection by nonconvergent methods," *Neural Networks*, Vol. 6, No. 6, pp. 771-783, 1993

Fischer, D.C., "Use of the ballistocardiogram to assess surgical risk," *Bibliotheca Cardiologica*, Vol. 27, pp. 21-24, 1971

Fisher, R., "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, Vol. 7, pp. 179-188, 1936

Friesen, G.M., Jannett, T.D., Jadallah, M.A., Yates, S.L., and Quint, S.R., "A comparison of the noise sensitivity of nine QRS detection algorithms," *IEEE Transactions on Biomedical Engineering*, Vol. 37, No. 1, pp. 85-98, 1990

Geva, S. and Sitte, J., "Adaptive nearest neighbor pattern classification," *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, pp. 222-318, 1991

Ghosh, J., Deuser, L.M., and Beck, S.D., "A neural network based hybrid system for detection, characterization, and classification of short-duration oceanic signals," *IEEE Journal of Oceanic Engineering*, Vol. 17, No. 4, pp. 351-363, 1992

Giles L.C. and Omlin, C.W., "Pruning recurrent neural networks for improved generalization performance," *IEEE Transactions on Neural Networks*, Vol. 5, No. 5, pp. 848-851, 1994

Gillis, P.S., Guzowki, T.S., Keller, B.L., and Kerr, R.H., "Test methodologies and design automation for IBM ASICs," *IBM J. Res. Develop.*, Vol. 40, No. 4, pp. 461-473, 1996

Gokhale, M., Holmes, W., Kopser, A., Lucas, S., Minnich, R., and Sweely, D., "Building and using a highly parallel programmable logic array," *Computer*, Vol. 24, No. 1, pp. 81-89, 1991

Gordon, J.W., " On certain molar movements of the human body produced by the circulation," *J. Anat. Physiol.*, Vol. 11, pp. 533-536, 1877

Grossberg, S., "Embedding field: a theory of learning with physiological implications," *Journal of Mathematical Psychology*, Vol. 6, pp. 209-239, 1969

Hamilton, W.F., Remington, J.W., Dow, P., "Relationship between cardiac ejection curve and ballistocardiographic force," *Am. J. Physio.*, Vol. 144, pp. 557, 1945

Hammamoto, Y., Uchimura, S., and Tomita, S., "On the behavior of artificial neural network classifiers in high-dimensional spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 5, pp. 571-574, 1996

Hammerstrom, D., "A VLSI architecture for high performance, low-cost, on-chip learning," in *Proceedings of International Joint Conference on Neural Networks*, pp. 537-543, 1990

Hanka, R., "Statistical pattern recognition applied to the analysis of ballistocardiograms," *IEEE Trans. on Biom.. Eng.*, Vol. 26, pp. 253, 1979

Harrison, R.F., Marshall, S.J., and Kennedy, R.L., "The early diagnosis of heart attacks: A neurocomputational approach," *in Proceedings of IEEE International Conference on Neural Networks*, pp. 1-5, 1991

Harrison, W.K., Talbot, S.A., and Baltimore, "Discrimination of the Quantitative Ultralow-frequency Ballistocardiogram in Coronary Heart Disease," *American Heart Journal*, Vol. 74, pp. 80-87, 1967

Hartman, E.J., Keeler, J.D., and Kowalski, J.M., "Layered neural networks with Gaussian hidden units as universal approximators," *Neural Computation*, Vol. 2, No. 2, pp. 210-215, 1990

Hasegawa, A., Itoh, K., and Ichioka, Y., "Generalization of shift invariant neural networks: Image processing of corneal endothelium," *Neural Networks*, Vol. 9, No. 2, pp. 345-356, 1996

Hassoun, M.H., Wang, D., and Spitzer, A.R., "NNERVE: Neural network extraction of repetitive vectors of electromyography-Part I: Algorithm," *IEEE Transactions on Biomedical Engineering*, Vol. 41, No. 11, pp. 1039-1052, 1994

Hertz, J., Krogh, A., and Palmer, R.G., "*Introduction to the Theory of Neural Computation*," Addison-Wesley, 1991

Hikawa, H., "Implementation of simplified multilayer neural networks with on-line learning," in *Proceedings of IEEE International Conference on Neural Networks*, pp. 1633-1636, 1995

Hjorth, B., "EEG analysis based on time domain properties," *Electroencephalogram Clin. Neurophysiol*, Vol. 29, pp. 306-310, 1970

Hoffbeck, J.P. and Landgrebe, D.A., "Covariance matrix estimation and classification with limited training data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 7, pp. 763-767, 1996

Hollis, P.W., Harper, J.S., and Paulos, J.J., "The effects of precision constraints in a backpropagation learning network," *Neural Computation*, Vol. 2, pp. 363-373, 1990

Holmstrom, L and Koistinen, "Using additive noise in backpropagation training," *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, pp. 24-38, 1992

Holt, J.L. and Hwang, J.N., "Finite precision error analysis of neural network hardware implementations," *IEEE Transactions on Neural Networks*, Vol. 42, No. 3, pp. 281-290, 1993

Hopfield, J.J., "Neural networks and physical systems with emergent collective computational abilities," in *Proceedings of the National Academy of Sciences*, pp. 2554-2558, 1982

Hornik, K., Stinchcombe, M., and White, H., "Multilayer feedforward networks are universal approximators," *Neural Networks*, Vol. 2, pp. 359-366, 1989

Hu, Y.H., Tompkins, W.J., Urrusti, J.L., and Afonso, V.X., "Applications of artificial neural networks for ECG signal detection and classification," *Journal of Electrocadiology*, Vol. 26, pp. 66-73, 1994

Huang, X., Liu, W., and Wei, W.Y., "A high-performance CMOS redundant binary multiplication-and-accumulation (MAC) unit," *IEEE Transactions on Circuits and System-I*, Vol. 41, No. 1, pp. 33-39, 1994

Humpert, B.K., "Improving back propagation with a new error function," *Neural Networks*, Vol. 7, No. 8, pp. 1191-1192, 1994

Hunt, S.D. and Deller, J.R., "Selective training of feedforward artificial neural networks using matrix perturbation theory," *Neural Networks*, Vol. 8, No. 6, pp. 931-944, 1995

Hurdle, J., Josephson, L., and Brunvand, E., "Reliable interfacing of self-timed FPGA-based neural classifiers," in *More FPGAs*, W. Moore and W. Luk, Eds., Oxford, pp. 377-386, 1993

Hwang, K., "*Computer Arithmetic: Principles, Architecture, and Design*," John Wiley & Sons, 1979

Jabri, M. and Flower, B., "Weight perturbation: An optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks," *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, pp. 154-157, 1992

Jacobs, R.A., "Increased rates of convergence through learning rate adaptation," *Neural Networks*, Vol. 1, pp. 295-307, 1988

Jansen, B.H. and Shankar, K., "Sleep stage with movement-related signals," *International Journal of Biomedical Computation*, Vol., 32, pp. 289-297, 1993

Jim, K., Giles, C.L., and Horne, B.G., "An analysis of noise in recurrent neural networks: Convergence and generalization," *IEEE Transactions on Neural Networks*, Vol. 7, No. 6, pp. 1424-1438, 1996

Joliffe, I.T., "*Principal Component Analysis*," Springer-Verlog Press, 1986

Karhunen, J. and Joutsensalo, J., "Generalizations of Principal Component Analysis, optimization problems, and neural networks," *Neural Networks*, Vol. 8, No. 4, pp. 549-562, 1995

Karhunen, J. and Joutsensalo, J., "Learning of robust principal component subspace," *in Proceedings of the International Joint Conference on Neural Networks*, pp. 2409-2412, 1993

Kiessling, C.E., "Preliminary appraisal of the prognostic value of Ballistocardiography," *Bibl. Cardiol.*, Vol. 26, pp. 292-295, 1970

Kittler, J. and Young, P., "A new approach to feature selection based on the Karhunen-Loeve Expansion," *Pattern Recognition*, Vol. 5, pp. 335-352, 1979

Kohonen, T., "An adaptive associative memory principle," *IEEE Transactions on Computers*, Vol. C-23, pp. 444-445, 1974

Kohonen, T., "*Self-Organization and Associative Memory*," Springer-Verlag, 1989

Kotilainen, P., Saarinen, J., and Kaski, K., "Hardware implementations of PCA neural networks," in *Artificial Neural Networks*, Aleksander, I. and Taylor, J., Eds., Vol. 2, Amsterdam, pp. 1427-1430, 1992

Krogh, A. and Hertz, J.A., "A simple weight decay can improve generalization," *in Advance in Neural Information Processing Systems 4*, Moody, J., Hanson, S., and Lippman, R., Eds., Morgan Kaufman Publishers, pp. 950-957, 1992

Kung, S.Y., "*Digital Neural Networks*," PTR Prentice Hall, 1993

Lee, C. and Landgrebe, D.A., "Decision boundary feature extraction for nonparametric classification," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 2, pp. 433-444, 1993

Lee, J.C., Shue, B.J., Fang, W.C., and Chellappa, R., "VLSI neuroprocessors for video motion detection," *IEEE Transactions on Neural Networks*, Vol. 4, No. 2, pp. 178-190, 1993

Lee, J.Y., Garvin, H.L., and Slayman, C.W., "A high-speed high-density silicon 8×8-bit parallel multiplier," *IEEE Journal of Solid-State Circuits*, Vol. SC-22, No. 1, pp. 35-40, 1987

Lee, S.W., "Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 6, pp. 648-652, 1996

Lemarie, P.G. and Meyer, Y., "Ondelettes et bases Hilbertiennes," *Revista Matematica Lbero Americana*, Vol. 2, 1986

Leong, P.W. and Jabri, M.A., "A low-power VLSI arrhythmia classifier," *IEEE Transactions on Neural Networks*, Vol. 6, pp. 1435-1445, 1995

Leung, S., Luk, A., and Ng, S., "Fast convergent genetic-type search for multi-layer network," *IEICE Trans. Fundamentals*, Vol. E77-A, No. 9, pp. 1484-1492, 1994

Lippmann, R.P., "Pattern classification using neural networks," *IEEE Commun. Mag.*, Vol. pp. 47-64, 1989

Lisa, F., Carrabina, J., Perez-Vicente, C., Avellana, N., and Valderrama, E., "Two bit weights are enough to solve vehicle license recognition problem," *in Proceedings of IEEE International Conference on Neural Networks*, pp. 1242-1246, 1993

Liu, W., Chen, S., and Cavin, R., "A bit-serial VLSI architecture for generating Moments in real-time," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 2, pp. 539-546, 1993

Lont, J.B. and Guggenbuhl, W., "Analog CMOS implementation of a multilayer peceptron with nonlinear synapses," *IEEE Transactions on Neural Networks*, Vol. 3, No. 3, pp. 457-465, 1992

Lovell, B.C. and Bradley, A.P., "The multiscale classifier," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 2, pp. 124-137, 1996

Lu, F. and Samueli, H., "A 200-MHz CMOS pipelined multiplier-accumulator using a quasi-domino dynamic full-adder cell design," *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 2, pp. 123-132, 1993

Lynn, T.N. and Wolf, S., "The prognostic significance of the ballistocardiogram in ischemic heart disease," *Am. Heart, J.*, Vol. 88, pp. 277-280, 1974

Maeda, Y., Hirano, H., and Kanata, Y., "A learning rule of neural networks via simultaneous perturbation and its hardware implementation," *Neural Networks*, Vol. 8, No. 2, pp. 251-259, 1995

Mallat, S.G., "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 674-693, 1989

Manzzoul, M.A. and Jayabharathi, D., "FPGA for fuzzy controllers," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 25, No. 1, pp. 213-216, 1995

Mao, J. and Jain, A.K., "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Transactions on Neural Networks*, Vol. 6, No. 2, pp. 296-317, 1995

Marchesi, M., Orlandi, G., Piazza, F., and Uncini, A., "Fast neural networks without multipliers," *IEEE Transactions on Neural Networks*, Vol. 4, No. 1, 1993

Marineli, R.A., Penney, D.G., Marineli, W.A., and Baciewicz, F.A., "Rotary motion in the heart and blood vessels: A review," *Journal of Applied Cardiology*, Vol. 6, pp. 421-431, 1991

Marr, D., "A theory for the cerebellar cortex," *Journal of physiology*, Vol. 202, pp. 437-470, 1969

Martin, C.E., Rogers, S.K., and Ruck, D.W., "Neural networks Bayesian error estimation," in *Proceedings of IEEE International Conference on Neural Networks*, pp. 305-308, 1994

Masters, T., "*Practical Neural Network Recipes in* C++," Academic Press, 1993

Matsuoka K. and Kawamoto, M., "A neural network that self-organizes to perform three operations related to principal component analysis," *Neural Networks*, Vol. 7, No. 5, pp. 753-765, 1994

McCulloch, W.S. and Pitts, W., "A logical calculus of ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, Vol. 5, pp. 115-133, 1943

Minsky, M. and Papert, S., "*Perceptrons*," MIT Press, Cambridge, 1969

Miyanaga, Y. and Tochinai, K., "Parallel VLSI architecture for multi-layer self-organizing cellular network," *IEICE Trans. Electron.*, Vol. E76-C, No. 7, pp. 1174-1181, 1993

Moini, A., Eshraghian, K., and Bouzerdoum, A., "The impact of VLSI technologies on neural networks," in *Proceedings of IEEE International Conference on Neural Networks*, pp. 158-163, 1995

Moon, T. and Mereny, E., "Classification of hyperspectral images using wavelet transforms and neural networks," in *Wavelet Application in Signal and Image Processing III*, Laine, A.F. and Unser, M.A., Eds., SPIES Press., pp. 725-735, 1995

Moor, R.K., "New systems and architectures for automatic speech recognition and synthesis," *Nato ASI*, France, 1984

Muller, U.A., Gunzinger, A., and Guggenbuhl, W., "Fast neural net simulation with a DSP processor array," *IEEE Transactions on Neural Networks*, Vol. 6, No. 1, pp. 203-213, 1995

Nagasaka, Y. and Iwata, A., "Data compression of long time ECG recording using BP and PCA neural networks," *IEICE Trans. Inf. & Syst.*, Vol. E76-D, No. 12, pp. 1434-1442, 1993

Nakayama, K., Inomata, S., and Takeuchi, Y., "A digital multilayer neural network with limited binary expressions," in *Proceedings of International Conference on Neural Networks*, pp. 587-592, 1990

Neale, R., "CPLD/FPGA happenings," *Electronic Engineering*, September, pp. 7, 1996

Nekovei, R. and Sun, Y., "Back-propagation network and its configuration for blood vessel detection in angiograms," *IEEE Transactions on Neural Networks*, Vol. 6, No. 1, pp. 64-71, 1995

Neubauer, A., "Robust learning algorithms for multilayer perceptrons with discretized synaptic weights," in *Proceedings of IEEE International Conference on Neural Networks*, pp. 3170-3174, 1995

New, B., "A distributed arithmetic approach to designing scaleable DSP chips," *EDN*, No. 8, pp. 107-114, 1995

Ney, H., Essen, U., and Kneser, R., "On the estimation of 'small' probabilities by leaving-one-out," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 12, pp. 1203-1212, 1995

Ng, K. and Lippman, R.P., "A comparative study of the practical characteristics of neural network and conventional pattern classifiers," *Advances in Neural Information Processing System-III*, pp. 970-975, 1990

Nichani, S. and Ranganathan, N., "Design of a systolic VLSI chip for computing scale space," in *Proceedings of Applications of Artificial Intelligence, Machine Vision and Robotics*, pp. 146-156, 1992

Noordergraaf, A. and Spier, D.H., "A new method for the elimination of the respiration effect in Ultra-Low frequency displacement and velocity ballistocardiograms," *in Proceedings of 3rd Eur. Symp. Ballistocardiography*, pp. 142-145, 1962

Noordergraaf, A., "Computers and the Ballistocardiogram," in *Proc. 1st World Conger. Ballistocard. Cardiovasc. Dynamics*, pp. 111-122, 1965

Noore, A., "Microcontroller compatible clock chip design using field programmable gate array," *IEEE Transactions on Consumer and Electronics*, Vol. 37, No. 3, pp. 629-634, 1991

Nowlan, S.J. and Hinton, G.E., "Simplifying neural networks by soft weight-sharing," *Neural Computation*, Vol. 4, No. 4, pp. 473-493, 1992

Oja, E., "A simplified neuron model as a principal component analyzer," *Journal of Mathematical Biology*, Vol. 15, pp. 267-273, 1982

Oja, E., Ogawa, H., and Wangviwattana, J., "Learning in nonlinear constrained Hebbian network," in *Artificial Neural Networks*, Kohonen, T., Ed., pp. 385-390, 1991

Oja, E., Ogawa, H., and Wangviwattana, J., "Principal component analysis by homogeneous neural networks, Part I & Part II: The Weighted Subspace Criterion," *IEICE Trans. INF. & Syst.*, Vol. E75-D, No. 3, pp. 366-381, 1992

Onodera, H. and Takeshita, K., "Hardware architecture for Kohonen network," *IEICE Trans, Electron.*, Vol. E76-C, No. 7, pp. 1159-1165, 1993

Palmieri, F. and Zhu, J., "Self-association and Hebbian learning in linear neural networks," *IEEE Transactions on Neural Networks*, Vol. 6, No. 5, pp. 1165-1184, 1995

Peterson, G.E., Clair, D.C., Aylward, R.A., and Bond, W.E., "Using Taguchi's method of experimental design to control errors in layered perceptrons," *IEEE Transactions on Neural Networks*, Vol. 6, No. 4, pp. 949-960, 1995

Petrosian, A., Homan, R., Pemmaraju, S., and Mitra, S., "Wavelet based texture analysis of EEG signal for prediction of epileptic seizure," *Wavelet Application in Signal and Image processing III*, Laine, A.F. and Unser, M.A., Eds., SPIE Press, pp. 189-194, 1995

Piche, S.W., "The selection of weight accuracy for Madalines," *IEEE Transactions on Neural Networks*, Vol. 6, No., 2, pp. 432-445, 1995

Plaut, D.C. and Hinton, G.E., "Learning sets of filters using back-propagation", *Computer Speech and Language*, Vol. 2, pp. 35-61, 1987

Plutowski, M. and White, H., "Selecting concise training sets from clean data," *IEEE Transactions on Neural Networks*, Vol. 4, No. 2, pp. 305-318, 1993

Polo, O., "*Partial Upper Airway Obstruction During Sleep Studies with the Static Charge-Sensitive Bed (SCSB)*," Helsinki University Press, Finland, 1992

Polo, O., Tafti, M., Hamalainen, M.H., Vaahtoranta, K., and Alihanka, J, "Respiratory variation of the ballistocardiogram during increased respiratory load and voluntary central apnea," *Eur. Respir. J.*, Vol. 5, pp. 257-262, 1992

Pynsent, P.B. and Hanka, R., "A simple program for a phaseless recursive digital filter," *Journal of Biomedical Eng.*, Vol. 4, pp. 252-253, 1982

Ramanan, S., Petersen, R.S., Clarkson, T.G., and Taylor, J.G., "pRAM nets for detection of small targets in sequences of infra-red images," *Neural Networks*, Vol. 8, No. 7/8, pp. 1227-1237, 1995

Raudys, S.J. and Jain, A.K., "Small sample size effects in statistical pattern recognition: recommendations for practitioners," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, pp. 252-264, 1991

Ream, A.K, "Advances in Medical Instrumentation," *Science*, Vol. 200, pp. 959-964, 1978

Reed, R., Marks, R.J., and Oh, S., "Similarities of error regularization, sigmoid gain scaling, target smoothing, and training with jitters," *IEEE Transactions on Neural Networks*, Vol. 6, No. 3, pp. 529-538, 1995

Reyneri, L.M. and Filippi, E., "An analysis of the performance of silicon implementations of backpropagation algorithms for artificial neural networks," *IEEE Transactions on Neural Networks*, Vol. 40, No. 12, pp. 1380-1389, 1991

Richez, J., "Top of R-wave trigger circuit," *Medical & Biological Engineering & Computing*, Vol. 30, No. 2, pp. 251-252, 1992

Rioul, O. and Vetterli, M., "Wavelets and signal processing," *IEEE Signals Processing Magazine*, October, pp. 14-38, 1991

Rogers, S.K., Colombi, J.M., Martin, C.E., Gainey, J.C., Fielding, K.H., Burns, T.J., Ruck, D.W., Kabrisky, M., and Oxley, M., "Neural networks for automatic target recognition," *Neural Networks*, Vol. 8, No. 7/8, pp. 1152-1184, 1995

Rohen, V.E., "*Application of Statistical Pattern Recognition Techniques to the Analysis of Ballistocardiograms*," Ph.D Thesis, University of Cambridge, Cambridge, U.K., 1985

Rosenblatt, F., "*Principles of Neurodynamics*, " Spartan, New York, 1962

Rosenblatt, F., "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, Vol. 65, pp. 386-408, 1958

Roy, A., Govil, S., and Miranda, R., "An algorithm to generate radial basis function (RBF)-like nets for classification problems," *Neural Networks*, Vol. 8, No. 2, pp. 179-201, 1995

Rubner, J. and Tavan, P., "A self-organizing network for principal component analysis," *Europhysics Lett.*, Vol. 10, pp. 693-698, 1989

Rumelhart, D.E., Hinton, G.E., and Williams, R.J., "Learning representations by backpropagating errors," *Nature*, Vol. 323, No. 9, pp. 533-536, 1986A

Rumelhart, D.E., Hinton, G.E., and Williams, R.J., "*Parallel Distributed Processing*," MIT Press, 1986B

Sakaue, S., Kohda, T., Yamamoto, H., Maruno, S., and Shimeki, Y., "Reduction of required precision bits for backpropagation applied to pattern recognition," *IEEE Transactions on Neural Networks*, Vol. 4, No. 2, pp. 270-275, 1993

Salomon, R. and Hemmen, J.L.V., "Accelerating backpropagation through dynamic self-adaptation," *Neural Networks*, Vol. 9, No. 4, pp. 589-601, 1996

Sanger, T.D., "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, Vol. 2, pp. 459-473, 1989

Schwerdt, H., Bette., L., Grillmaier, R., Hoffmann, W., Schieffer, H.T., and Stager, H.K., "Design and evaluation of a new computer based force vector ballistocardiograph," *Med. & Biol. Eng. & Comput.*, Vol. 25, pp. 453-458, 1987

Shin, Y. and Ghosh, J., "Ridge polynomial networks," *IEEE Transactions on Neural Networks*, Vol. 6, No. 3, pp. 610-622, 1995

Shukla, D.V., Belina, J., Gonzalez, O.L., Razvi, J.H., and Dodhia, S., "Wavelet transform for small dimension neural network pattern classification of subtly different ECGs," in *Proceedings of 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 729-730, 1993

Sietsma, J. and Dow, R.J.F., "Creating artificial neural networks that generalize," *Neural Networks*, Vol. 4, No. 1, pp. 67-79, 1988

Sietsma, J. and Dow, R.J.F., "Neural net pruning-why and how," *in Proceedings of IEEE International Conference on Neural Networks*, pp. 325-333, 1988

Smith, M., "*Neural Networks for Statistical Modeling*," Van Nostrand Reinhold, 1993

Smith, N.T., "Ballistocardiography," in *Noninvasive Cardiology*, Weissler, A.M., Ed., Grune & Stratton, 1974

Solis, F.J. and Wets, J.B., "Minimization by random search techniques," *Mathematics of Operations Research*, Vol. 6, pp. 19-30, 1981

Song, H.H. and Lee, S.W., "LVQ combined with simulated annealing for optimal design of large set reference models," *Neural Networks*, Vol. 9, No. 2, pp. 329-336, 1996

Stamkopoulos, T., Strintzis, M., and Maglaveras, N., "One-lead ischaemia detection using a new backpropagation algorithm and the European ST-T Database," *in Proceedings of IEEE International Conference of Computers in Cardiology*, pp. 663-666, 1992

Starr, I. and Noordergraaf, A., "*Ballistocardiography in Cardiovascular Research*," Philadelphia, Lippincott, 1967

Starr, I. and Wood, F.C., "Twenty years studies with ballistocardiography, the relation between the amplitude of the first record of 'healthy' adults and eventual mortality and morbidity from heart disease," *Circulation*, Vol. 23, pp. 714-732, 1961

Starr, I., Rawson, A.J., Schroeder, H.A., and Joseph, N., R., "Studies on the estimation of cardiac output in man; and of abnormalities in cardiac function from the heart's recoil and the blood impact", *Am. J. Physiol.*, Vol. 127, pp. 1-28, 1939

Stevenson, M, Winter, R., and Widrow, B., "Sensitivity of feedforward neural networks to weight errors," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, pp. 71-80, 1990

Sudjianto, A. and Hassoun, M.H, "Statistical basis of nonlinear Hebbian learning and application to clustering," *Neural Networks*, Vol. 8, No. 5, pp. 707-715, 1995

Szu, H., Telfer, B., and Garcia, J., "Wavelet transforms and neural networks for compression and recognition," *Neural Networks*, Vol. 9, No. 4, pp. 695-708, 1996

Talbot S.A., Harrison, W.K., Ginn, W.M., "Features of ULF-BCG pertinent to coronary heart disease," in *Ballistocardiography Cardiovascular Dynamics*, Knoop, A.A., Ed., Basel, pp. 55-68, 1966

Taylor, W.K., "Electrical simulation of some nervous system functional activities," in *Information Theory*, Cherry, C., Ed., Butterworths, pp. 314-328, 1959

Thakor, N.V., Sun, Y., Rix, H., and Caminal, P., "Multiwave: a wavelet-based ECG data compression algorithm," *IECE Trans. Inf. & Syst.*, Vol. E76-D, No. 12, pp. 1462-1468, 1993

Tollencere, T., "Super SAB: fast adaptive back propagation with good scaling properties," *Neural Networks*, Vol. 3, pp. 561-573, 1989

Troyer, T.W., "Feedforward Hebbian learning with nonlinear output units: A Lyapunov approach," *Neural Networks*, Vol. 9, No. 2, pp. 321-328, 1996

Viewlogic, "*Viewlogic Reference Manual,*", Viewlogic System Inc., Massachusetts, 1993

Voft, L. Silny, J., Rau, G., and Effert, S., "A pattern recognition system for interpretation of Electrocardiographic body surface potential maps," in *Proceedings of IEEE International Conference of computer in Cardiology*, pp. 73-76, 1992

Watanabe, T., Kimura, K., and Aoki, M., "A single 1.5-V digital chip for a 10^6 synapse neural networks," *IEEE Transactions on Neural Networks*, Vol. 4, No. 3, pp. 387-393, 1993

Weideman, W.E., Manry, M.T., Yau, H., and Gong, W., "Comparisons of a neural network and a nearest-neighbor classifier via the numeric hand print recognition problem," *IEEE Transactions on Neural Networks*, Vol. 6, No. 6, pp. 1524-1530, 1995

Weigend, A.S., Rumelhart, D.E., and Huberman, B.A., "Generalization by weight-elimination applied to currency exchange rate prediction," *in Proceedings of International Joint Conference on Neural Networks*, pp. 837-841, 1991

Werbos, P., "*Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*," Ph.D. Thesis, Harvard University, USA, 1974

Widrow, B. and Winter, R., "Neural nets for adaptive filtering and adaptive pattern recognition," *IEEE Computer*, Vol. 3, pp. 25-39, 1988

Widrow, B., "Generalization and information storage in networks of Adaline 'Neurons'," in *Self Organizing Systems*, Yovits, M.C., Jacobi, G.T., and Goldstein, G.D., Eds., pp. 435-461, 1962

Wiesspeiner, G. and Xu, W., "Multichannel ambulatory monitoring of circulation related biosignals," in *Proceedings of IEEE International Conference of Computers in Cardiology*, pp. 457-460, 1992

Wright, B.M., "A Portable Ballistocardiograph," *Bibl. Cardiol.*, Vol. 33, pp. 73-75, 1975

Xilinx, "*The Programmable Logic Data Book*", Xilinx Inc., California, 1994

Xilinx, "*XACT Reference Guide*", Xilinx Inc., California, 1993

Xu, L. and Yuille, A.L., "Robust principal component analysis by self-organizing rules based on statistical physics approach," *IEEE Transactions on Neural Networks*, Vol. 6, No. 1, pp. 131-143, 1995

Xu, L., "Least mean square error reconstruction principle for self-organizing neural-nets," *Neural Networks*, Vol. 6, pp. 627-648, 1993

Yamakashi, K., Kuroda, M., Tanuka, S., Yamaguchi, I., and Kawarada, A., "Non-conscious and automatic acquisition of body and excreta weight together with ballistocardiogram in a lavatory," To appear in *Proceedings of 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE Press, 1996

Yasunaga, M., Masuda, N., Yagyu, M., Asai, M. et al., "A self-learning digital neural network using wafe-scale LSI," *IEEE Journal of Solid-State Circuits*, Vol. 28, No. 2, pp. 106-113, 1993

Yoo, H., and Pimmel, R.L., "The effect of weight precision and range on neural network classifier performance," *Neurocomputing*, Vol. 6, pp. 541-549, 1994

Yu, X, Dent, D., and Osborn, C., "Classification of Ballistocardiography using wavelet transform and neural networks," to appear in *Proceedings of 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE Press, 1996

Yu, X., and Dent, D., "Implementing neural networks in FPGAs," in *Proceedings of IEE Colloquium on 'Hardware Implementations of Neural Networks and Fuzzy Logic'*, London, pp. 7/1-7/5, 1994

Yu, X., Dent, D., and Osborn, C., "Hardware requirements for a neural-based BCG classifier prototype," in *Proceedings of International Workshop on Medical & Biological Signal Processing*, pp. 182-188, 1995

Yu, X., Dent, D., and Osborn, C., "The selection of weight precision for ballistocardiography classification," in *Proceedings of IEEE International Conference on Neural Networks*, pp. 2485-2489, 1995

Yu, X., Gong, D., Osborn, C., and Dent, D., "A wavelet multiresolution and neural network system for BCG signal classification," in *Proceedings of IEEE Region Ten Conference on Digital Signal Processing Applications*, pp. 491-495, 1996

Yu, X.H. and Chen, G.A., "On the local minima free condition of backpropagation learning," *IEEE Transactions on Neural Networks*, Vol. 6, No. 5, pp. 1300-1303, 1995

Zhao, G., "*Development of New Ballistocardiographic System and Applications of Neural Networks to the Analysis of Ballistocardiograms*", Ph.D. Thesis, University of Cambridge, Cambridge, U.K., 1994

Appendix A

Deriving Filter Coefficients

The following is extracted from the method for computing the coefficients for the filter H corresponding to the scaling function $\phi(x)$ applied by Mallat (1989) based on polynomial spline functions. The frequency response of the filter $H(\omega)$ is related to the filter coefficients $h(n)$ by the Fourier transform:

$$H(\omega) = \sum_{n=-\infty}^{+\infty} h(n)e^{-in\omega} \qquad \text{(A.1)}$$

The Fourier transform of a scaling function of $\phi(x)$ is defined as $\tilde{\phi}(\omega)$. In Mallat's original paper (1989), he has proved that the $\tilde{\phi}(\omega)$ is related to the Fourier transfer function $H(\omega)$ of a quadrature mirror filter as follows:

$$\tilde{\phi}(2\omega) = H(\omega)\tilde{\phi}(\omega) \qquad \text{(A.2)}$$

The Fourier transform of the scaling function associated with multiresolution approximations built from polynomial splines of order $2p+1$ is given by Lemarie and Meyer (1986) as follows:

$$\tilde{\phi}(\omega) = \frac{1}{\omega^n \sqrt{\sum_{2n}(\omega)}} \qquad \text{(A.3)}$$

where $n = 2 + 2p$ and the function $\sum_n(\omega)$ is defined as

$$\sum_n(\omega) = \sum_{k=-\infty}^{+\infty} \frac{1}{(\omega + 2k\pi)^n} \qquad \text{(A.4)}$$

As suggested by Mallat (1989), one can compute a closed form of $\sum_n(\omega)$ by calculating the derivative of order $n-2$ of the equation:

$$\sum_2(\omega) = \frac{1}{4 \cdot \sin^2(\omega/2)} \qquad \text{(A.5)}$$

From A.2 and A.3, one can get

$$H(\omega) = \sqrt{\frac{\sum_{2n}(\omega)}{2^{2n}\sum_{2n}(2\omega)}} \tag{A.6}$$

For the case $p = 1$ and thus $n = 4$, the multiresolution approximation is built from cubic splines. By inverse Fourier transform, one can obtain the filter coefficients $h(n)$ correspond to the impulse response of H. The first 12 filter coefficients as computed by Mallat (1989) are shown in Table A.1. The corresponding scale function $\phi(x)$ is shown in Figure A.1.

Table A.1: Filter Coefficients (Mallat, 1989)

h(0)	h(1)	h(2)	h(3)	h(4)	h(5)	h(6)	h(7)	h(8)	h(9)	h(10)	h(11)
0.542	0.307	-0.035	-0.078	0.023	-0.03	0.012	-0.013	0.006	0.006	-0.003	-0.002

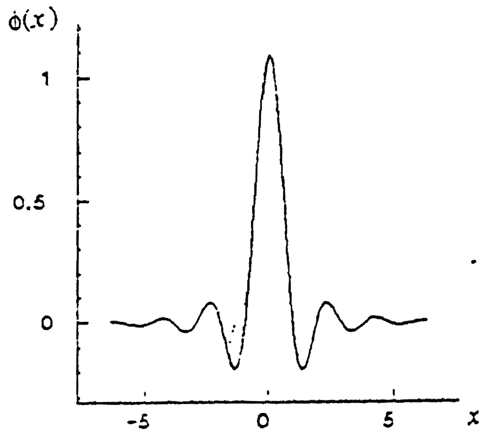


Figure A.1: Wavelet scaling function $\phi(x)$ (Mallat, 1989).

Appendix B

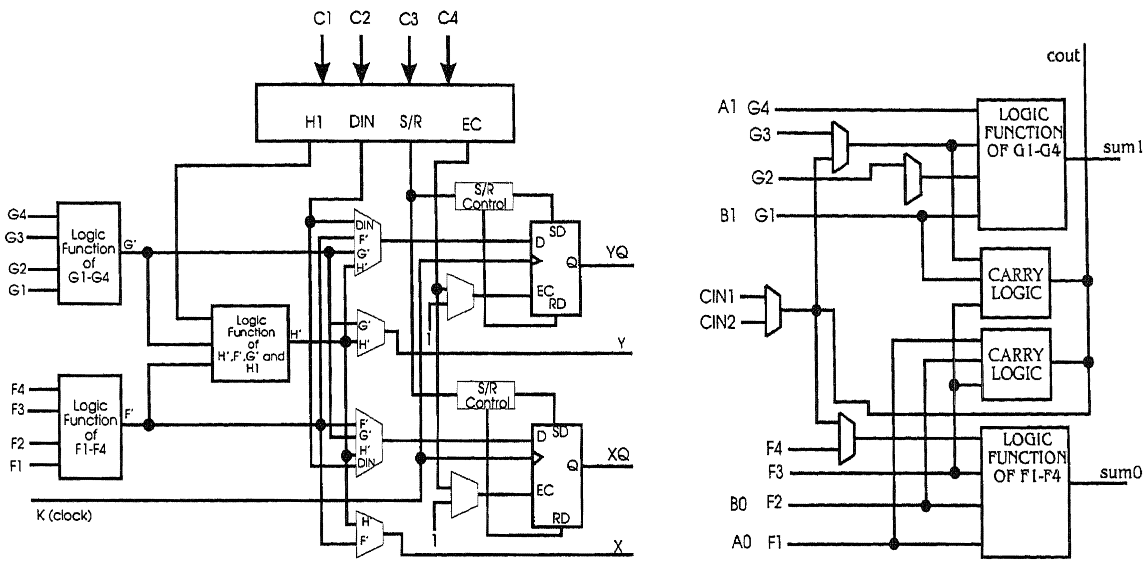
Xilinx XC4000 Series

The first generation XC2000 family was introduced in 1985. This family has two members ranging in complexity from 800 to 1500 gates. In 1987, the second generation XC3000 devices were introduce with a complexity range from 1,300 to 10,000 usable gates. In 1990, the third generation XC4000 family devices , with on-chip memory capability, featured logic densities up to 25,000 usable gates. The XC4000 family devices also have a fast carry propagate adder feature which provides dedicated circuitry for the computation and interconnection of carries. To further reduce the chip cost, XC5200 family devices were introduced in 1995. The XC5200 family has the same fast carry logic feature and high density as the XC4000 family, but no on-chip RAM is available on the chip. However, the cost for the XC5200 family is only half of the XC4000 series. In 1996, the new XC6200 FPGA architecture was introduced that is specifically designed for implementing reconfigurable coprocessors in embedded system applications. The XC6200 architecture features a built-in processor interface and support for fast partial or full reconfiguration (Xilinx, 1995).

Table B.1: XC4000 Family Overview

	4003	4005	4006	4008	4010	4013	4020	4025
Usable Gates	3,000	5,000	6,000	8,000	10,000	13,000	20,000	25,000
Max. RAM Bits	3,200	6,272	8,192	10,368	12,800	18,432	25,088	32,768
No. of CLBs	100	196	256	324	400	576	784	1,024
Max I/Os	80	112	128	144	160	192	224	256

Table B.1 shows for the XC4000 family of FPGAs that the maximum number of usable gates is 25,000. The architecture of the XC4000 family consists of three types of configurable elements: a perimeter of I/O blocks (IOBs); a core of Configurable Logic Block (CLBs); and an interconnection area. The CLBs provide the functional elements for constructing the user's logic. The IOBs provide the interface between the package pins and internal signal lines. The programmable interconnect resources provide routing paths to interconnect the inputs and outputs of CLBs and IOBs. Customised configuration is established by programming internal static memory cells that determine the logic functions and the interconnections implemented in the logic cell array (LCA). Figure B.1 shows the block diagram of a single CLB. Each CLB has two independent 4-input function generators (F1 - F4 and G1 - G4). These function generators, whose outputs are labelled F' and G', are each capable of implementing any arbitrarily defined Boolean function of their four inputs as memory look-up tables. A third function generator, H', can implement any Boolean function of its three inputs: F' and G' and a third input from an outside block (H1). Thus, a CLB can be used to implement any two independent functions of up to four variables, or any single function of five variables. The two storage elements in the CLB are edge-triggered D-type flip-flops with common clock (K) and clock enable (EC) inputs. A third common input (S/R) can be programmed as either an asynchronous set or reset signal independently for each of the two registers (Xilinx, 1994).



Simplified Block Diagram of XC4000 Configurable Logic Block

Fast Carry Logic in Each CLB

Figure B.1 Simplifier Diagram of an XC4000 CLB and the fast carry logic embedded in each CLB

In addition, each CLB F' and G' function generator contains dedicated arithmetic logic for the fast generation of a carry signal. The two 4-input function generators can be configured as a 2-bit adder with built-in hidden carry that can be expanded to any length. This dedicated carry circuitry opens the door to many new applications, where the previous generations of FPGAs were not fast and/or not efficient enough. It greatly increases the efficiency and performance of arithmetic unit design.

Appendix C

List of the Original Communications

In this appendix, the author's published work in which they have been referenced in this thesis are included. The papers are arranged in time sequence.

Yu X. and Dent D. [1994]. Implementing neural networks in FPGAs. IEE Colloquium on 'Hardware Implementations of Neural Networks and Fuzzy Logic. 7/1-7/5

Yu X., Dent D. and Osborn C. [1995]. Hardware requirements for a neural-based BCG classifier prototype. Proc. International Workshop on Medical & Biological Signal Processing. 182-188

Yu X., Dent D. and Osborn C. [1995]. The selection of weight precision for Ballistocardiography classification. Proc. IEEE International Conference on Neural Networks. 2485-2489

Yu X., Dent D. and Osborn C. [1996]. Classification of Ballistocardiography using wavelet transform and neural networks. Proc. 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 626

Yu X., Gong D., Osborn C. and Dent D. [1996]. A wavelet multiresolution and neural network system for BCG signal analysis. Proc. IEEE TENCON Digital Signal Processing Applications. 491-495

IMPLEMENTING NEURAL NETWORKS IN FPGAs

*Xinsheng Yu and Don Dent**

ABSTRACT

The reconfigurability of certain field programmable gate arrays(FPGAs) has shown their advantage of flexibility in digital system design. With the availability of greater density and high speed of FPGAs, the ability to realise special purpose processors will become possible. In this paper, we present our research work of implementing specific trained neural network in Xilinx XC4000 series FPGAs for portable digital system prototype in heart disease classification process.

INTRODUCTION

Artificial neural networks(ANNs) can solve difficult problems in areas such as pattern recognition, image processing, and medical diagnostic[1-2]. The biologically inspired ANN models require the massive parallel computation. Thus, the high speed operation in real time applications can be achieved only if the networks are implemented using parallel hardware architecture.

Implementing ANNs is the procedure of mapping the neural model into hardware in an efficient form. Research in this area falls into two categories: general-purpose neurocomputers for emulating a wide range of neural network models and special purpose VLSI neural systems which are dedicated to a specific neural network model. The general purpose neurocomputers offers the flexibility and a high degree of observbility into the inner working of neural algorithms, and the special purpose VLSI designs provides high speed and compactness for typical applications. One can find a wide variety of implementation studies and techniques in these area [3-7].

The special-purpose VLSI chips allow to overcome neural network implementation problems in real time applications with enhanced processing speed and compactness, but due to different neural models and limited use it has been prohibitively expensive and time consuming to develop such chips. Such VLSI chips are fixed mapping, so the degree of flexibility is limited. With the introduction of field programmable gate arrays(FPGAs), it is feasible to provide custom hardware for application specific computations design. The changes in designs in FPGAs can be accomplished within a few hours and thus result in significant savings in cost and design cycle. Cox et. al. [6] proposed a method of implementing a fully connected feedforward network with Xilinx FPGAs for image processing that the single processing node was partitioned into two XC3090 chips. Ruckert et. al. [7] reported a neural associative memories implementation based on RAMs and XC3090 FPGAs. However, it is desirable to improve the speed and size in an FPGA based design for portable system implementations. This paper deals with efficient implementation of feedforward networks in table-lookup FPGAs for a portable digital system in classification of balliscardiogram(BCG) for automatic heart disease monitoring. By making use of the features of XC4000 series FPGAs and low precision method, we can significantly reduce the number of CLBs in hardware implementation for compact design.

NEURAL NETWORKS HARDWARE

In the neural model, each neuron computes a weighted sum of its inputs and applies a non-linear function to its results. Each input is either tied to the output of another neuron or to an external input. Architectural parameters, such as the number of inputs per neuron, and each neuron's conductivity vary considerably within a network, and from application to application. A special-purpose neural network hardware must be carefully balanced for the special nature and efficient implementation within reach of today's technology.

---

* Faculty of Design & Technology, University of Luton, Park Square, Luton LU1 3JU

We can distinguish two phases of operation in many neural network applications. During the learning phase, the topology and weights of the network are determined from a labelled set of examples using a rule such as the backpropagation algorithm. In the subsequent retrieval or classification phase, the network recognises pattern based on information stored in the architecture and weights during training. Since the computational and infrastructure requirements during the learning phase are considerably more complex than classification, it is considered that the efficient implementation of network is to separate the learning and retrieval phase in hardware. The network parameters can be determined during learning phase off the line. After learning, these parameters are downloaded into hardware processors specialised for the classification task.

Precision of processing elements have a significant impact on area, power, and performance. The efficient hardware implementation can be achieved by matching suitable arithmetic resolution. Experiments and theoretical study show that the precision requirements of neurons within a single network vary[8]. Many research works show that during the training procedure the weights are very sensitive to the precision and number range used. Once the networks is fully trained, the number of bits needed for weights can be very low and remains virtually unchanged in a classification task[9]. Thus the hardware complexity can be further optimised by matching the computational accuracy of the processor to the requirements of typical neural networks to simplify the hardware design complexity and improve the processing speed.

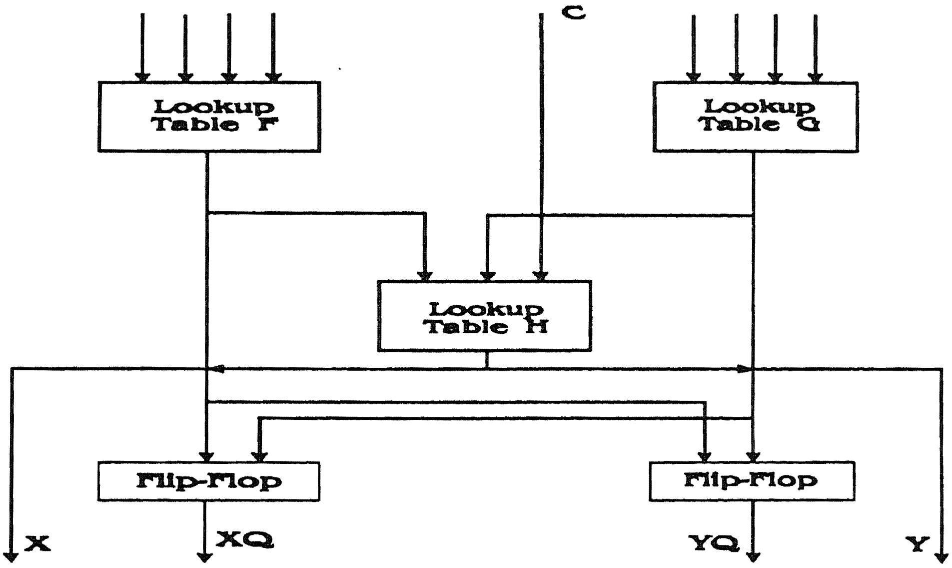


Figure 1. Simplifier Diagram of an XC4008 CLB

XILINX FPGAS OVERVIEW

The Xilinx's XC4008 products were adopted in our study, as the new Xilinx 4000 series offer the improved architecture and short time delays. The Xilinx 4000 series of FPGAs uses lookup tables to implement combination logic. The density of XC4008 is in 8,000 equivalent gates and the device consists of a 18×18 grid of configurable logic blocks(CLBs) interconnected horizontally and vertically by programmable routing lines. Several long lines provide a reduced delay of large fanouts for each row and column of interconnects. In the speed grad-6 XC4008 the minimum interconnect delay 1.9 ns occurs between adjacent CLBs. Interconnection delay increase with increasing fanout and routing distance. In a simplified diagram, the configuration of each CLB is shown in Figure 2. The throughput delay of each CLB is shown in Table 1.

The XC4008 also has a fast carry propagate adder feature which provides dedicated circuitry for the computation and interconnection of carries. The fast propagate addition feature provides an implementation of the least delay and area[10].

Configuration	Delay (ns)
Table F or G to Output X or Y	6.0
Tables F and/or G through H to Output X or Y	8.0
Flip Flop Input to Output(Setup and Clock-to-Q)	5.0
C inputs via H to X or Y	7.0

Table 1. Throughput Delay of a CLB

NETWORK ARCHITECTURE BASE ON FPGAS

Usually the most popularly used neural model in classification application is feedforward networks with backpropagation learning rule. The neurons in the network are arranged in layers, each of which receives inputs only from neurons in the previous layer. The basic operation performed by a neuron during classification is a weighted sum, followed by a non-linear transform function f:

$$y = f(\sum_i x_i w_i + b)$$

Where y is the output of node, x_i is referred to the input of the neuron and w_i is the parameters of the weights. Each input is either tied to the output y of another neuron or to an external input. Optionally, a bias b may be added to the weighted sum. In our application a sigmoid function $f = (1 + e^{-(\sum x_i w_i + b)})^{-1}$ is used.

When mapping algorithm to FPGAs, it is desirable to exploit the maximum amount of parallelism for the trade off area and for incremental speedup. Instead of direct calculation, the sigmoid function is implemented in a lookup table.

The three layers network is trained by off-line simulation of the network on a PC with backpropagation rule. The extracted BCG signal features are normalised as inputs with 7-bit. First the network is trained in normal high resolution calculation, and then gradually reduce the number of precision bits. Finally the weights are carefully quantized to approximately 4 bits. The implementation of the fully connected network is mapped on to the Xilinx FPGA. The same layer processing nodes parallelism is adopted in our architecture. The input layer nodes are just simply buffer the input data.

The architecture of a single hidden processing node is given in Figure 2. The multiplication is an 7-bit variable multiplied by an 5-bit constant which produces an 12-bit product. The special fixed multiplier architecture[6] is used. Each bit of the 12-bit result can be expressed as a function of 7-bit variable. As the XC4000 series CLB can implement the functions of up to nine variables, one CLB can generates 1-bit of the 12-bit product. The 12-bit product will require 12 CLBs. For XC3000 series only have 5 variable inputs per CLB, it need to break the multiplication into two 4-bit by 5-bit operations. One addition process is required to sum the two partial products. By making use of XC4000 series our multiplier architecture directly produces the final product and does not need the extra addition operation. Therefore, the delay of partial product addition cycles are reduced.

ARCHITECTURE	MULTIPLIER	ADDER
XC3000	8 CLBs(partial multiplication)	30 CLBs
XC4000	12 CLBs	9 CLBs

Table 2 Comparison of XC3000 and XC4000 implementation

Our experiment shows that a factor of 4 to input 12-bit is enough to prevent the overflow for the summing operation. The carry lookahead adder is used for the adder design. For this 16-bit adder the 30 CLBs are required for XC3000 series with 50 ns delay, but by making of use the fast carry logic of XC4000 series only 9 CLBs are needed with 20.5 ns performance. In Table 2 we give the CLBs required for multiplication and adder implementation.

As the 16-bit product-sum is not effective due to the saturation characteristic of the sigmoid function, a scaling process is used to scale the sum into 10 bit value to address the activation function lookup table. The 10 bits address and 7 bits output data will require about 1 Kx7-bit PROM or RAM.

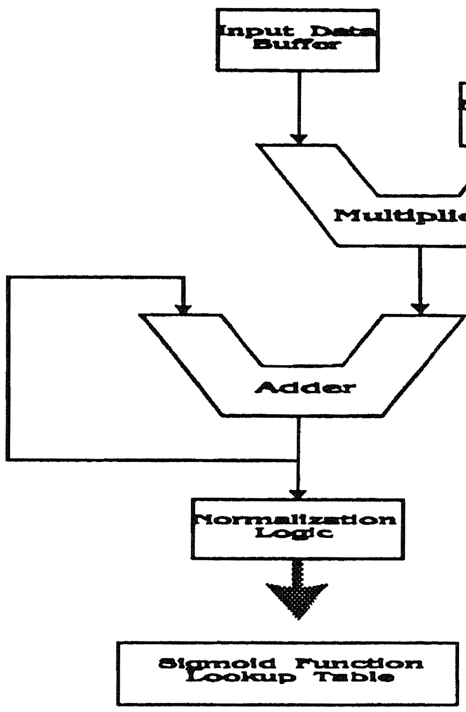


Figure 2. A block diagram of single hidden node implementation

DISCUSSION AND CONCLUSION

The goal of our work is to implement a neural based instrument for automatic heart disease screening. It is desirable that the parameters of hardware network can be easily altered when the patients database improved later. As FPGAs allow the hardware design via configuration software control, the improvement of circuitry design is just a matter of modifying, debugging and downloading the new configuration code in a short time. There are many mature and powerful CAD tool technology to support the FPGAs design. It is possible to incorporate user-defined function into standard libraries. To design a circuitry, which is to perform a given function and which is to be implemented on a FPGAs, instances of appropriate functional modules are invoked from the library and then interconnected. The FPGAs approach offers the very desirable features such as design flexibility, expandable size and low cost for rapid circuit prototyping.

In classification area, the necessary calculation accuracy varies by application. The low precision can simplify the hardware implementation complexity and speedup the performance. With the availability of recent high density FPGAs, such as the available XC4013 which the equivalent gates is up to 13,000, it is possible to include more processing nodes and circuitry in a single chip for the moderate size network implementation.

with available XC4000 series the same arithmetic unit implementation can reduce nearly half of number CLBs required in XC3000 series.

In summary, the FPGAs approach for network implementation provides the flexibility as programmable system. For the neural based instrument prototype in real time application, conventional specific VLSI neural chips design suffer the limitation in time and cost. This problem can be overcome with configurable FPGAs techniques. The specific parameters can be configured to allow the adaptation to a variety of difference classification tasks. With low precision arithmetic design, we can achieve adequately high speed and small size for real time pattern classifier implementation. The programmability of the configurable FPGAs yields the availability of fast special purpose hardware for wide applications. Its programmability could set the conditions to explore new neural network algorithms and problems of a scale that would not be feasible with conventional processors.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the valuable discussion with C. Osborn and Dr. D. Crecraft. This work is supported by the University grant.

REFERENCES

1. S. Farrugia, H. Yee, and P. Nickolls, "Implantable Cardioverter Defibrillator Electrogram Recognition with a Multilayer Perceptron," PACE, Vol. 16, no. 1, pp 228-234, 1993
2. M. Poliac, J. Zanetti, and D. Salerno, "Performance Measurements of Seismocardiogram Interpretation Using Neural Networks," Computer in Cardiology, pp573-576, IEEE Computer Society, 1992
3. U. Ramache, "SYNAPSE: A Neurocomputer that Synthesize Neural Algorithms on a Parallel Systolic Engine," Journal of Parallel and Distributed Computing, Vol. 14, pp 306-318, 1992
4. T. Watanabe, M. Aoki, K. Kimura, T. Sakata, and K. Itoh, "The Advantages of a DRAM-Based Digital Architecture for Low-Power, Large-Scale neuro-Chip," IECE Trans. Electron, Vol. E76-c, No. 7, pp 1206-1214, 1993
5. J. Burr, "Digital Neurochip Design," in 'Parallel Digital Implementations of Neural Networks,' Prentice-Hall Inc., pp 223-282, 1993
6. C. Cox and W. Blanz, "GABGLION-A Fast Field-Programmable Gate Array Implementation of a Connectionist Classifier," IEEE Journal of Solid-state Circuits, Vol. 27, No. 3, pp288-299, 1992
7. U. Ruckert, A. Funke and C. Pintaske, "Acceleratorboard for Neural Associative Memories," Neurocomputing, Vol.5, No.1, pp 39-49, 1993
8. M. Stevenson, R. Weinter, and B. Widow, "Sensitivity of Feedforward Neural Networks to Weight Errors," IEEE Transactions on Neural Networks, Vol. 1, No. 2, pp 71-80, 1990
9. F. lisa, J. Carrabina, C. Vicente, N. Avellana, and E. Valderrama, "Two-bit Weights Are Enough to Solve Vechicle License Recognition Problem," Proc. IEEE International Conference on Neural Networks, pp. pp. 1242-1246, 1993
10. Xilinx, "The Programmable Logic Data Book," San Jose, 1993

Printed and published by the IEE, Savoy Place, London WC2R 0BL, UK.

HARDWARE REQUIREMENTS FOR A NEURAL-BASED BCG CLASSIFIER PROTOTYPE

Xinsheng Yu, Don Dent and Colin Osborn
School of Electronics and Computer Engineering, Faculty of Design and Technology, University of Luton

ABSTRACT

Artificial neural networks have been shown to be a practical solution in many medical and engineering applications. This leads to an increasing interest in specific hardware implementations of neural networks. This paper describes an empirical method to replace the sigmoid function with a threshold function for a limited weight precision network in a Ballistocardiogram classification application. The simulation results show that such networks have a good performance compared with high precision networks with sigmoid activation function. The advantage of this method is that the hardware implementation can be further reduced. An investigation involving the implementation of one hidden layer network with three input, three hidden and three output nodes in a field programmable gate arrays chip is presented.

INTRODUCTION

A ballistocardiogram (BCG) complex is the pattern of small body movements caused by the mechanical activity of the heart. The advantage of the BCG method is that no electronic leads need to be attached to the patient. The development of the ultra-low-frequency techniques has enabled further improvement in BCG research. The BCG has been used to diagnose coronary disease [1], to differentiate uncomplicated hypertension from hypertensive heart disease [2], and to diagnose congestive heart disease [3]. Various recording and analysis methods have been developed. Alihanka [4] developed a static charge sensitive bed (SCSB) used in the home for whole night sleep recording. This method enables long term monitoring of basic physiological functions such as cardiac function and respiration, together with their respective regulatory mechanisms, as well as of different motor activities in sleep. Hanka [5] proposed a method to record the short term BCG signal with a chair to screen people into normal, hypertension and heart attack risk categories. However, the reported BCG interpretation methods rely on tape recording followed by analysis with digital computers. The limitation of this processing technique is that it is expensive and inconvenient for a routine screening procedure. Thus, the research and development of an easy-to-use, small size but inexpensive real time system, which can automatically classify the BCG into categories, is desired to fill the gap.

The ability of neural networks to learn from examples has given powerful approaches in image processing, pattern classification and non-linear dynamic system modelling. The feed forward multilayer perceptron (MLP) model has been shown to provide practical solutions to a variety of problems in medical diagnostic applications. A feed forward neural network is one kind of parallel information processing system and the conventional serial machine implementation of such an algorithm cannot always meet real time process requirements. Therefore, the operation speed will be achieved by a parallel hardware implementation solution. On the other hand, some portable, stand-alone instruments require that the system will be ready to process real time information without complex programming. The topology and weights of the network can be determined off-line in a computer and, after learning, these parameters are downloaded into hardware processors specialised for the classification task. Therefore, we focus on this area of hardware implementation.

The research results show that weight precision has a significant impact on power, system cost and speed in neural hardware design [6]. The problem of selecting suitable precision for weights has been studied by theoretical [7] and empirical [8] methods. The theoretical approach could be used as a guideline to chose a suitable precision. However, this method does not find the optimal low precision in a specific application. Empirical methods can be used to find the lowest precision at the cost of extra training time. It is felt that little research has been carried out on the effects of low precision weights on accuracy in medical applications. Thus, an empirical approach to investigate the effects of low precision weights in a network for BCG classification has been carried out in our previous work [9].

This paper describes a method of replacing the sigmoid function with a threshold function for the limited weight precision networks to further simplify the network implementation. The performances of the networks were compared with our previous results. The possibility of implementing the trained network in field programmable gate arrays (FPGAs) has been investigated.

WHY FPGAs ?

Neural hardware has been studied over past decades. Most of the reported works are based on application specific integrated circuits (ASICs) which are targeted at specific algorithms. These platforms offer excellent performance and complexity, but they need specific software and hardware development tools, and a long time to validate the design which rarely makes a cost-effective solution at the system prototype stage. On the other hand, there is little market for small-to-medium general purpose neural chips in application-orientated neural circuits research.

FPGAs are a programmable platform and the gate arrays can be reprogrammed an unlimited number of times. With such technology, the function in the hardware can be reprogrammed and loaded with new algorithms under program control. Therefore, the changes in designs in FPGAs can be accomplished within a few hours which results in significant savings in cost in the design cycle. It is considered that FPGAs are ideal candidates in medical instrument prototyping. First, to develop truly robust ANN based classifiers, large data sets are required to train the network and to validate the results. It is expected that the network weights could be updated as the patients database increases. The reprogrammability of FPGAs in situ is suitable for this dynamic environment. Second, the cost is comparatively low, considering the time and fabrication charges of building custom ASIC circuits, especially for low volume and prototype products. Third, recent advances in semiconductor technology and improved architecture has made FPGAs available in a complexity which is suitable for ANN design, with many mature and powerful CAD tools supporting the FPGA development. Finally, research in the applications of ANNs is still an open area. It is desirable that neural circuitry has the ability to accommodate changes to new algorithms.

BCG SIGNAL AND FEATURE EXTRACTION

A one dimensional BCG signal is recorded from two ultra-low frequency acceleration transducers on the bottom of the chair. A single channel surface Electrocardiogram (ECG) is simultaneously recorded via the two arms of the chair. The R wave peak of the ECG signal is used to define the BCG beats. Both BCG and ECG signals are sampled at 100 samples/second with 8-bit resolution. An averaging technique is used to recover the useful information from background activity and system noise. Figure 1 shows a sequence of BCG signals with R wave trigger (Figure 1-A) and an averaged BCG signal (Figure 1-B). The averaged BCG signal is then normalised to give 80 samples in the time scale. A Kittler and Young linear transform as proposed by the University of Cambridge Clinical School is used to extract the features. Their analysis results show that three features were accurate enough to represent information for classification. Following our analysis, the coefficients of the system eigenvectors were rounded to 10-bit. The 80 inputs are transformed to a three dimensional feature vector with 18-bit resolution.

(A) (B)

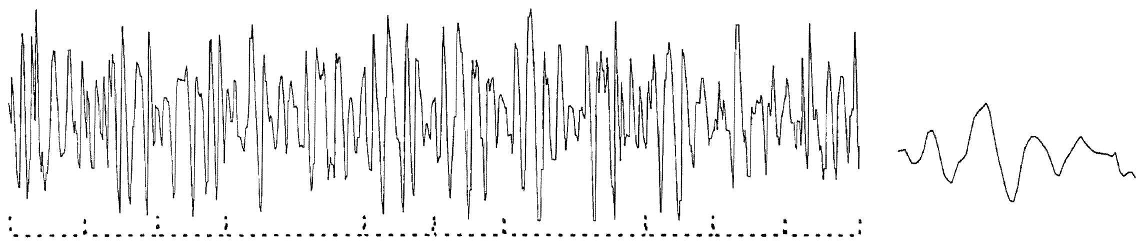


Figure 1: (A) BCG signal with R wave trigger (dash line) and (B) after average BCG signal

SIMULATION RESULTS

With BCG feature data sets of normal (59 subjects), hypertension (275 subjects) and heart attack risk (7 subjects) that were provided by the University of Cambridge, an experiment was conducted to investigate different weight precision effects on the classification accuracy. The standard backpropagation learning rule is adopted in our study. We divide the available data set into two parts, training and test. 49% of normal subjects, 40% of hypertension and 57% heart attack risk patients are used as a training data set. The remaining data are used as a test set to verify the network performance on patterns not seen during training. The sigmoid activation function $f(x) = 1/(1+\exp(-x))$ is used for the hidden and the output nodes. The target values of the output nodes for the training set are 0 and 1, respectively. In the evaluation phase, if the output value of an output node exceeded the threshold (0.5) then it was considered to be 1, and if it was less than the threshold (0.5) then it was considered to be 0. An output vector of [1, 0, 0] represented a normal subject, an output vector of [0, 1, 0] represented a hypertension subject, and an output [0, 0, 1] represented a heart attack risk subject. Output of [1, 1, 0], [0, 1, 1], [1, 1, 1], [1, 0, 1] and [0, 0, 0] were indeterminate(nondiagnostic). If the actual output does not match the desired output, e. g. for the normal subject the output is [0, 1, 0] or [0, 0, 1], it is considered to be a missclassification. The training was performed on a 486 based PC with single floating point precision.

The experiment tested the perceptron with varying numbers of hidden nodes, as shown in Table 1. As can be seen, there was little change in performance between 3 hidden nodes and 7 hidden nodes.

Table 1: Number of hidden nodes test

No. of Nodes	1	2	3	4	5	6	7
accuracy(%)	95.92	96.94	97.45	97.45	97.45	97.96	96.94

Several inputs precision were tested. The network was firstly trained with high resolution calculation calculations with full 18-bit input resolution. Then the input vectors were rounded to 7-bit and 4-bit respectively and the networks were re-trained. The result is shown in Table 2. The 7-bit input is used for the remainder of this study.

Table 2: Performance of input range test

	18-bit input		7-bit input		4-bit input	
	Training Data Set	Testing Data Set	Training Data Set	Testing Data Set	Training Data Set	Testing Data Set
Correct(%)	99..31	97.45	99.31	97.45	97.93	97.45
Incorrect(%)	0.69	2.04	0.69	2.04	2.07	2.04
Indeterminate(%)	0	0.5	0	0.5	0	0.5

In our previous study[9], we have developed a method for selecting limited precision weights for BCG classification. By using this method, the network was trained with high precision calculation first. After the network convergence, the weight values are normalised in a small range and the network is re-trained using same training set, until the network converges. Then the weight precision is reduced step by step during the training[8], until a network with satisfactory generalisation performance is obtained. This provides weight values in the range of $\pm 2^{(n-1)}$, where n-bit is used to represent the weight. The results obtained by this method is shown in Figure 2. Figure 2 (a) and (b) show the results of training data set. Figure 2 (c) and (d) are the results of testing data set.

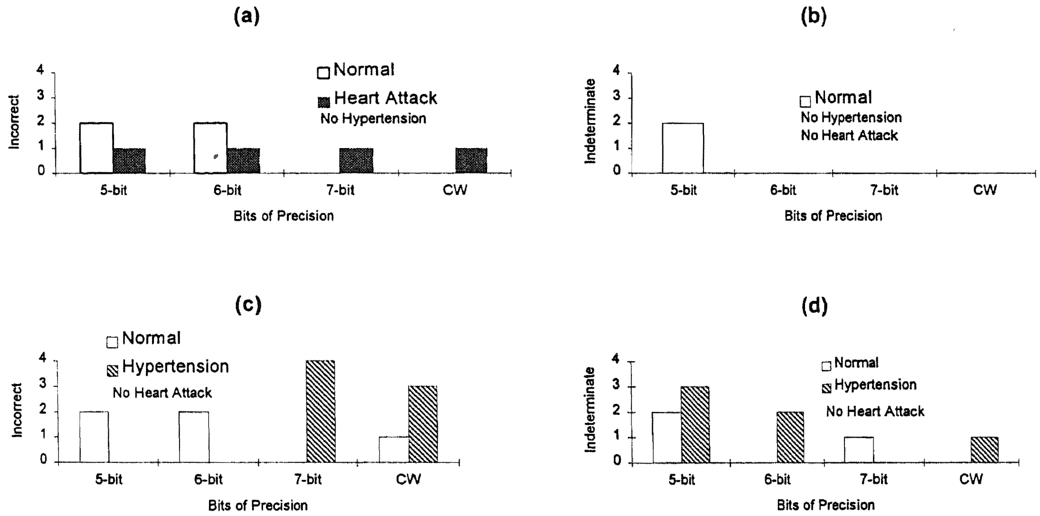


Figure 2: (a) and (b) show the number of incorrect and indeterminate classification for the training data test. (c) and (d) show the number of incorrect and indeterminate classification for the testing data set respectively.

Although this method has reduced the weight precision to 7-bit, a non-linear sigmoid function is still required which needs more memory to store a lookup table. Therefore, it is desirable that, if possible, the sigmoid function could be replaced by a simple threshold function while still keeping the limited weight precision expression. We modified the training algorithm to increase the gain parameter (λ) of the sigmoid function $f(x) = 1/(1+e^{-\lambda x})$ and train the network in a high precision calculation. When the network converged, λ was increased by one. The training process was repeated until the error was stable when further increasing λ. The method described previously was used to obtain low precision weights. Finally, the network weights are transferred to another networks of which the transfer function is a threshold function [0, 1]. The results are summarised in Figure 3. There is no indeterminate situation in threshold function networks. It indicates that a 7-bit weight network has a better accuracy in the testing data set and has one normal subject missclassified in the training data set compared with high precision sigmoid function networks. The 6-bit weight threshold network has the same performance in the training data set and a better accuracy in the testing data set. Thus, the 6-bit weight network is used in hardware implementation study.

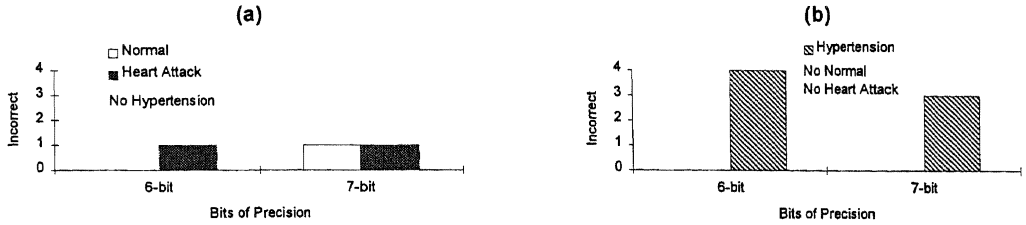


Figure 3: Threshold networks test results, (a) for training data set and (b) for testing data set.

NEURAL NETWORK IMPLEMENTATION

The Xilinx XC4000 series of FPGAs use lookup tables to implement combination logic. The configurable logic blocks (CLBs) are interconnected horizontally and vertically by programmable routing lines. Each CLB contains two lookup tables. A CLB can be used to implement any two independent functions of up to four variables, any single function of five variables, any function of four variables together with some functions of five variables, or it can implement even some functions of up to nine variables. They also have a fast carry propagate adder feature which provides dedicated circuitry for the computation and interconnection [10].

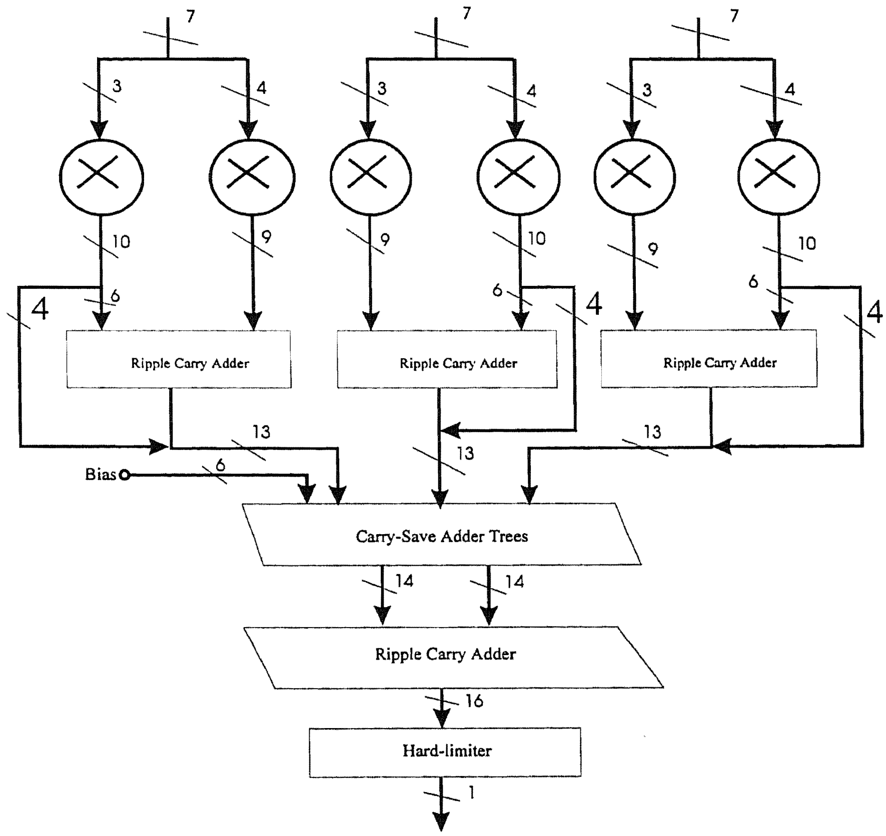


Figure 4: The block diagram of the hidden node implementation

The Xilinx XC4010 can be used for implementing the 6-bit weight precision network. The architecture of the XC4010 consists of 400 CLBs with a complexity of up to 10,000 usable gates. The input layer nodes just simply buffer the input data. For the hidden layer node, each input will be multiplied by its weight, and then the products will be summed together with a bias. If the final sum is bigger than 0, output a 1, otherwise, output a 0. The sign of the final sum can be used to generate the output. The Baugh-Wooley algorithm [11] is adopted to achieve two's-complement multiplication. The multiplication is a 7-bit variable(input feature) multiplier by an 6-bit signed constant (weight). Thus, the known weights are hardwired in the FPGA logic and the multiplication is broken into one 4-bit by 6-bit and one 3-bit by 6-bit multiplication with one addition [12]. As each CLBs can implement any two independent function of four variables, one CLB is used to generate two bits of the product. A ripple carry adder (RCA) is used to add the two partial products together. Three weighted input with the bias are added by using a carry save adder (CSA) tree of four inputs. A RCA is used to add the sum vector and the carry vector of the CSA into one final sum. To improve the throughput rate, a pipeline scheme can be incorporated in the design. The sign of the final sum is used to generate the threshold value 1 or 0. Figure 4 shows a function block diagram of a hidden nodes.

The output layer nodes design can be further simplified. As the input precision is only 1-bit, 1-bit by 6-bit multiplication is just the simple weight 'and' '1' or '0' operation. Thus, the partial products multiplication and summation can be eliminated. Furthermore, the weight value is known so the adding operation can be implemented in a lookup table which is similar to the multiplication implementation we mentioned above. The three outputs of the hidden nodes directly address the lookup table. Thus, the CLBs used for output nodes can be significantly reduced. A comparison of the CLBs required to implement the arithmetic units of sigmoid function and threshold function networks with 6-bit weight expression is summarised in Table 3

Table 3: Comparison of implementation of threshold function network and sigmoid function with FPGAs

	Sigmoid function	Threshold function
Multiplication	198 CLBs	99 CLBs
Summation	210 CLBs	117 CLBs
PROMs	3	0

The conventional FPGA base network implementation needs external memory to hold the sigmoid function table, such as, PROMs [12] or RAMs. However, with the method of replacing the common sigmoid function by a threshold function, the design complexity can be significantly reduced. It is possible to implement the network with a single XC4010 chip, which make the design more compact and improve the speed.

CONCLUSIONS AND FUTURE WORK

The simulation results suggest that, if limited precision weights can be found to produce the desired output in a common sigmoid function network, it is possible to replace the sigmoid function with a threshold function by gradually increasing the slope of the sigmoid function and re-training. It is interesting to note that, using the method described above, the networks have better classification accuracy for the testing data set than those common sigmoid function networks. As the data sets for the symptom of emergency heart attack subjects are limited, it is difficult to compare the classifier performance in this category. However, the results of normal and hypertension subjects have shown good agreement. Therefore, it is concluded that it is possible to use the threshold function to replace the non-linear transfer function in this application without a significant effect on classification accuracy. It demonstrates that this method has a great advantage in reducing the complexity of the hardware implementation.

An investigation of a threshold function network implementation has been presented. With this method, it is possible that the proposed network can be realised in a single XC4010 chip. Although the implementation example is a small network, it can be expand to realise more complex applications. To verify the classifier performance, further development of a large database is required and a theoretical investigation to exploit the property will be carried out.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. R. Hanka and his research team of the Clinical School of the University of Cambridge for their collaboration on this project. We thank Dr. D. Crecraft for his technical advice. We also thank Dr R. Harvey, Dr. R. Hearing and Mr J. Caplan for their encouragement and help. This programme was supported by the University of Luton.

REFERENCES

1. R. Brown and V. Delalla, "Ballistocardiogram, description and clinical use," Am. J. Med. Vol. 9, pp. 718-727, 1950
2. R. Brown, V. Dellala, M. A. Epstein and M. J. Hoffman, "clinical Ballistocardiography," The Macmillan Company , New York, 1952
3. Starr, "Clinical studies with ballistocardiograph: in congestive failure, on digitalis action, on change in ballistic form and in certain acute experiments," Am. J. M. Sc. 204, 469, 1941
4. Alihanka, "Basic principles for analysing and scoring Bio-Matt(SCSB) recordings," Annales Universitatis Turkuensis, D26, 1-129, 1987
5. Dr R. Hanka, private communication, 1993-1994
6. J. B. Burr, "Digital Network Implementations," in Neural Networks Concepts, Applications , and Implementations, Edited by P. Antognetti and V. Milutinovic, Prentice-Hall, Inc. 1991
7. L. Holt and J. Hwang, "Finite precision error analysis of neural network hardware implementations," IEEE Transactions on Computers, Vol. 42, No. 3, pp. 281-290, 1993
8. Nakayama, S. Inomata, and Y. Takeuchi, " A digital multilayer neural network with limited binary expression," Proc. Int. Joint. Conf. Neural Networks, Vol. 2, pp. 587-592, 1990
9. X. Yu, D. Dent and C. Osborn, "The selection of weights precision for Ballistocardiography classification," To appear in Proc. of IEEE ICNN'95, 1995
10. Xilinx, "The programmable logic data book," San Jose, USA, 1994
11. C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," IEEE Trans. Comput., Vol. C-22, pp. 1045-1047, 1973
12. E. Cox and W. E. Blanz, " Ganglion a fast field programmable gate array implementation of a connectionist classifier," IEEE Journal of Solid-State Circuits, Vol. 27, No. 3, pp. 288-299, 1992

The Selection of Weights Precision for Ballistocardiography Classification

Xinsheng Yu, Don Dent and Colin Osborn
Faculty of Design and Technology, University of Luton, Luton, LU1 3JU, England

ABSTRACT

Artificial Neural Networks have been effectively applied to a variety of medical diagnostic and classification situations. More recently, there is a growing interest in implementing such algorithms for real-time monitoring or fast-time scanning in the classification of normal and abnormal. These applications will benefit from hardware implementation. It is well known that the input and weights precision has a considerable impact on the circuit's complexity, speed and power consumption. This paper evaluates two training methods for selecting limited precision weights for Ballistocardiogram classification and suggests that a 7-bit weight precision can provide a similar performance compared with a high precision weights expression. Furthermore, a method is proposed for implementing the artificial neural networks using field programmable gate arrays for the rapid prototyping of algorithm specific hardware.

1. Introduction

The development of ultra-low-frequency techniques has enabled further interest in Ballistocardiography(BCG) research. The BCG is a movement related signal caused by shifts in the centre of mass of the blood that consists of components attributable to cardiac activity, respiration, and body movement. It has great potential value in that it directly reflects heart action and provides important diagnostic information. The BCG is different from other heart monitoring methods in that no electrodes need be attached to the human subjects. Therefore, it may be used as a low-cost tool for heart disease screening. However, most of the reported BCG interpretation methods rely on two separate procedures that is, tape recording followed by analysis with digital computers[1][2], which are cumbersome and inconvenient for routine screening procedures. Thus, research and development on a convenient, small size but inexpensive real time system, that can automatically classify the BCG into categories, is still a challenge.

Biologically inspired artificial neural networks(ANNs) potentially offer a compact, powerful approach for solving classification and prediction problems. ANNs have been successfully employed in a variety of medical diagnostic situations such liver cancer diagnosis[3], ECG signal analysis[4] and heart attack detection[5]. Many medical applications require a portable, stand-alone and easy to use classifier that can quickly process real time information. The system should be ready to operate without complex programming. This means that, after learning, the network parameters can be transferred to the hardware implementation to indicate the patient category. To this end, it is our goal to implement the parallel neural-like classification algorithms in hardware to efficiently process real time information.

One problem with the hardware implementation of neural networks is the multiplication of input vectors by synaptic weights. A large silicon area is required for even simple hardware multipliers. The reported work shows that the weight precision required to solve a particular application requirement depends on the problem difficulty and network architecture used[6]. As the relationship of the number of weight bits to circuit complexity, speed and cost is significant, the optimal weight size has been investigated so that the appropriate balance between diagnostic accuracy and hardware cost can be determined. On the other hand, in turning the design concepts into a applicable system, an instrument prototyping stage is usually required to evaluate the initial trial algorithms and to be demonstrated in field tests. In medical applications, it is also necessary that the implemented network has the ability to adapt as the patient database expands. Thus, a flexible efficient neural circuit is required to meet the need.

In this paper, the performance of different weight precision in a BCG classification application is evaluated with modified backpropagation and weighted error function learning algorithms. Using a a BCG database developed in the University of Cambridge Clinical School , an experiment was conducted to investigate the variations in the diagnostic accuracy with the precision of weights. The simulation tests were performed using normal, hypertension and heart attack risk cases. Finally a flexible hardware platform using field programmable gate arrays(FPGAs) is proposed for neural network implementation in the medical applications.

0-7803-2768-3/95/$4.00

2. BCG signal processing and feature extraction

A one dimensional BCG signal is obtained from two ultra-low frequency acceleration transducers on the bottom of a experiment chair constructed in the University of Cambridge Clinical School. A single channel surface Electrocardiogram(ECG) is simultaneously recorded via the two arms of the chair and the R wave peak of the ECG signal is used to define the BCG beats. Figure 1 shows the ECG and BCG signals. Both the BCG and ECG signals are sampled at 100 samples/second with 8-bit resolution. The 20 beats of the BCG signals are averaged to recover the useful information from background activity and system noise. The averaged BCG signal is then normalised to give 80 samples in the time scale. A Kittler and Young linear transform is used to select the features. From the analysed results produced by the University of Cambridge Clinical School, three features were chosen for this study. The feature vector is fed to a feed forward multilayer perceptron(MLP) to classify the feature vectors into associated categories.

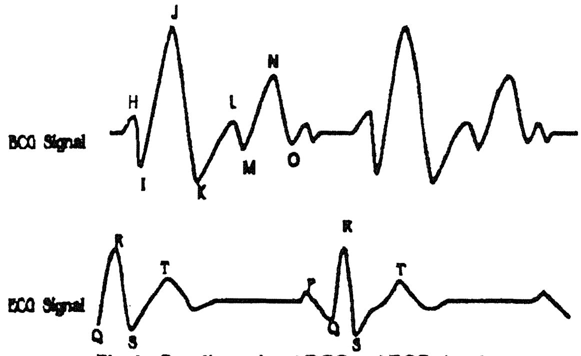


Fig. 1: One dimensional BCG and ECG signal.

3. Neural networks simulation with limited weight precision

The network is a feed forward, fully connected three-layer network with 3 input nodes, corresponding to the 3 dimensional feature vector. The output layer consists of three nodes representing people who have normal heart, or hypertension, or potential heart attack risks. The standard backpropagate(BP) rule[7] is used to train the network with varying hidden nodes from 1 to 9. The results show that, except for the network with 6 hidden nodes that has a slightly better performance than 3 hidden nodes, the performance of 3 hidden nodes has the same or better performance than all other networks. Therefore, the hidden nodes are fixed to 3 nodes considering the trade-off between performance requirement and ease hardware implementation.

The feature data sets used throughout this study are summarised in Table 1. All the feature vectors are normalised between 0 to 1. The initial weights are generated randomly. The target values of the training set are 0, 1 respectively. Several experiments were conducted to generate the random weights for training and then tested with test data sets. The highest accuracy initial values are used for later experiments. The activation function is a sigmoid function with range 0 to 1. Each output represented one of the three classes. An output larger than 0.5 was considered as 1, and an output smaller than 0.5 was considered as 0. An output vector of [1, 0, 0] represented a normal subject, an output vector of [0, 1, 0] represented a hypertension subject, and an output [0, 0, 1] represented a heart attack risk subject. Output of [1, 1, 0], [0, 1, 1], [1, 1, 1], [1, 0, 1] and [0, 0, 0] were indeterminate(nondiagnostic). If the actual output does not match the desired output, e.g. for the normal subject the output is [0, 1, 0] or [0, 0, 1], it is considered to be a missclassification.

Table 1: Training and Testing Data Sets

Symptom	Training Data Set	Testing Data Set	Total
Normal	30	29	59
Hypertension	111	164	275
Heart Attack Risk	4	3	7

3.1 Network training

In order to test what precision of input signal can preserve the required information, an experiment was conducted to test different input precision representations. The results showed that 7-bit input precision is sufficient to represent the features information for classification. Therefore, the input signals are fixed to 7-bit.

The weights precision required for the conventional BP method has been studied and error analysis shows that 12-bit to 14-bit precision is required to learn successfully over a range of applications[6]. Nakayama et al. [8] proposed an algorithm that gradually reduces the number of precision bits as the learning process continuous. This method demonstrates an effective way to reduce the weights precision to 2-bit for digital numeral patterns recognition. However, the results are obtained from a input of binary. Sakaue et al.[9] presented a weighted error function(WEF) learning algorithm and the results show that it has a better performance than conventional BP in 12-bit and 8-bit precision training. In this study, we modified the conventional BP rule in a similar way to that proposed by Nakayama. We refer this learning method as the modified BP(MBP) learning rule. The MBP method is compared with the WEF algorithm in weight precision selection. When the training had finished, both network performances were measured on the test data set.

All simulations were performed on a personal computer(PC) with floating point or fixed point forms respectively. With the MBP method, the network was trained with continuous weight(CW) in high resolution calculation using the entire training set. The conventional iterative gradient descent method is used to minimise the mean square error between the desired output and the actual output from the net which is defined as:

$$E = \sum_k (t_k - o_k)^2$$

where E is the error, t_k is the desired output, and o_k is actual node output.

We start by training the network with high precision calculation. After the network convergence, the weight values are normalised in a small range and the network is re-trained using same training set, until network is converged. Then the weight precision is reduced step by step during the training, until a network with satisfactory generalisation performance is obtained. The weight values are now in the range $\pm 2^{(n-1)}$, where n-bit are used to represent each weight.

The WEF algorithm is different from the conventional BP method in that the mean square error is minimised by using a weight coefficient g_k and a skipping coefficient G. Thus, the minimised square error function is rewritten as:

$$E = G\sum \{g_k(t_k - o_k)^2\}$$

The g_k and G are defined as the following

$$g_k \begin{cases} = m > 1 & if |o_k - t_k| > T_g \\ = 1 & otherwise \end{cases}$$

$$G \begin{cases} = 0 & if \forall_k |o_k - t_k| < T_G \\ = 1 & otherwise \end{cases}$$

During training, the weights are expressed as the q-bit integer part and the s-bit fractional part. If the absolute value is greater than $2^q - 2^{-s}$, then it will be rounded to $2^q - 2^{-s}$; values less than 2^{-s} are rounded to 2^{-s}.

4. Results and discussion

The test data set was used to test the trained network. All weights are expressed as n-bit, which includes 1 signed bit. Table 2 shows the results by using the WEF training method and Table 3 illustrates the results of MBP method.

Table 2: Performance of WEF method

Weight	12-bit	8-bit
Correct(%)	97.45	96.43
Incorrect(%)	1.02	1.02
Indeterminate(%)	1.53	2.55

Table 3: Performance of MBP method

Weight	CW	7-bit	6-bit	5-bit
Correct(%)	97.45	97.45	97.96	96.43
Incorrect(%)	2.04	2.04	1.02	1.02
Indeterminate(%)	0.5	0.5	1.02	2.55

The results in Table 2 indicate that 12-bit weight error function training algorithm gives the same result as the BP method with the CW expression. When training the network with 8-bit weights, this method increased the indeterminate rate and the accuracy becomes low. This is believed to be caused by inadequate weight update information. The results in Table 3 showed that 7-bit weight precision can give the same performance as continuous weights. The 6-bit weight network gave the highest accuracy. When further reducing the weight precision to 5-bit, the accuracy is decreased dramatically and the indeterminate rate gets higher. This could be caused by inadequate weight precision to keep the required information.

The performance were further assessed by calculating sensitivity and specificity[10], which are given in Table 4 and Table 5 for both methods.

Table 4: Sensitivity(%)

	WEF		MBP			
	12 bits	8 bits	CW	7-bit	6-bit	5-bit
Normal	96.55	93.1	96.55	100	93.1	93.1
Hypert-ension	98.17	98.17	97.56	97.56	98.78	99.39
Heart Attack	100	100	100	100	100	100

Table 5: Specificity(%)

	WEF		MBP			
	12-bit	8-bit	CW	7-bit	6-bit	5-bit
Normal	99.4	98.8	98.8	98.2	100	100
Hypert-ension	96.88	96.88	96.88	96.88	93.75	87.5
Heart Attack	99.48	99.48	99.95	99.95	100	98.96

The ideal requirement in medical diagnosis is that both sensitivity and specificity approach 100%. The comparison of these results indicates that the 12-bit weights precision obtained from the WEF training and 7-bit weights obtained from the MBP training have equal or better performance than the high resolution BP trained network in terms of sensitivity and specificity. However, the preliminary results showed the MBP method is more effective to obtain the lowest precision.

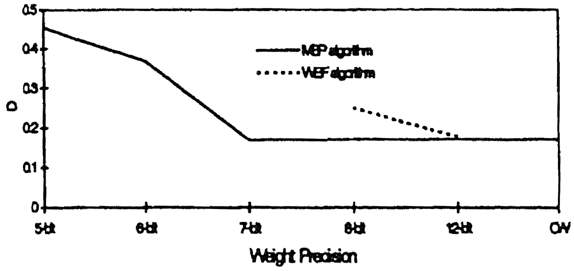


Fig. 2: Average value of hidden node derivative for different weight precision

Furthermore, the robustness of the networks was evaluated, in case reducing weight precision made them sensitive to small perturbations. Murray et. al[11] indicated that hidden nodes that are firmly ON or OFF are less likely to change state as a result of small variations in the input data and it is reasonable to infer that this will improve the generalisation ability of a classifier. This matter is also confirmed in Nakayama's study[8]. Therefore, we adopt Murray's method to measure the hidden layer stability by calculating the average of the output activation derivative over the whole test data set:

$$D = \frac{1}{P}\sum_{P}[\frac{1}{N}\sum_{N} O_h(1-O_h)]$$

where P is the number of patterns, N is the number of hidden nodes, and O_h is the hidden node activation output. The results are shown in Fig. 2 and confirm that the 7-bit weight precision network trained with MBP methods has a similar value of D compared with the high resolution network. The WEF method provides nearly the same stability in 12-bit precision, but the averaged value is increased when the weight precision is decreased.

5. Implementing a trained network with FPGAs

Recently, there is an increasing number of very large scale integration(VLSI) neural networks and one can find some implementations based on FPGAs[12],[13]. FPGAs are a programmable platform and the gate arrays can be reprogrammed an unlimited number of times. With such technology, the function in the hardware can be reprogrammed and loaded with new algorithms under program control. It is considered that they are an ideal platform in medical instrument prototyping in terms of performance, design cycle and cost.

The previous reports of network implementation in FPGAs are based on the Xilinx XC3000 series which have some limitations because of their architecture. The XC4000 family provides a regular, flexible programmable architecture of configurable logic block(CLBs), interconnected by a powerful hierarchy of versatile routing resources and surrounded by a perimeter of programmable input/output blocks(IOBs)[14]. Each CLB includes high-speed carry logic that can be activated by configuration. A CLB can be used to implement any two independent functions of up-to-four variables, or any single function of five variables, or any function of four variables together with some functions of five variables, or it can implement even some functions of up to nine variables. The XC4010 has been used in our implementation study. The architecture of XC4010 consists of 400 CLBs with a complexity of up to 10,000 usable gates.

The multiplication operation is a 7-bit variable(input) multiplied by an 7-bit signed constant(weight). Thus, the known weights can be hardwired in the FPGA logic and the multiplication can be broken into 4-bit by 7-bit and 3-bit by 7-bit multiplication with one addition[12]. One CLB is used to generate two bits of the products. Therefore, one synaptic multiplier only requires 11 CLBs. As the XC4000 series provide the fast carry logic, the partial products adder can be implemented in 5 CLBs. Pipelined carry save adder(CSA) trees are used to implement the summing operation. A ripple carry adder merges the sum vector and the carry vector of the CSA into one final sum. Our simulation shows that the significant 10-bit of the product-sum is adequate to address the sigmoid look up table due to the saturation characteristic of the sigmoid function. Thus, the final sum is scaled to 10-bit value to address the sigmoid function look up table which is implemented in a PROM. Therefore, three hidden nodes with a bias can be realised in a single XC4010 chip with 3 external PROM. The fully parallel network can be implemented in two XC4010 chips with three PROMs.

6. Conclusions

In this paper, the preliminary results indicate that a limited binary weight expression can still provide a satisfactory performance compared with a high precision weights expression. The WEF algorithm of 12-bit weight precision can perform a 12-bit precision learning and provide similar performance to a high precision network. It is shown that the performance of a 7-bit precision weight network trained with the MBP method can give the same performance as a high precision weight network in hypertension and heart attack risk categories. The simulation results suggests that 5-bit precision for weights is the minimum requirement in this specific application. It is also demonstrated that the application of low precision network produced results that are both accurate and robust. It is noted that the database used in this study is insufficient especially for the symptom of heart attack. However, the results of normal and hypertension subjects support the general conclusion.

The results of this study show that it is feasible to reduce the precision bits required in the BCG classifier without significantly reducing the accuracy. This simplifies the hardware implementation complexity and increases the processing speed. The possibilities for the realisation of trained neural networks in FPGAs was studied. With such technology, the function in the hardware can be changed and loaded within a few hours which results in significant savings in cost in the design cycle. It is shown that the approach using FPGAs for network implementation provides the flexibility for instrument prototyping in real time applications.

7. Acknowledgement

The authors would like to thank Dr R. Hanka from the University of Cambridge Clinical School for providing the BCG data sets and technical advice. We thank Dr C. Crecraft for valuable discussion. Thanks also to Dr R. Harvey, Dr. R Hearing and Mr J. Caplan for their encouragement and support. This research program was supported by the University of Luton.

8. References

[1] H. Jansen and K. Shankar, "Sleep staging with movement-related signals," International Journal of Biomedical Computing, pp. 289-297, Vol. 32, 1993

[2] O. Polo, M. Tafti, M. Hamalainen, K. Vaahtoranta and J. Alihanka, "Respiratory variation of the ballistocardiogram during increased respiratory load and voluntary central apnoea," Eur. Respir. Journal, pp. 257-262, Vol. 5, 1992

[3] L. Astion and P. Wilding, "Application of neural networks to interpretation of laboratory data in cancer diagnosis," Clinical Chemistry, pp. 34-38, Vol. 38, No. 1, 1993

[4] S. Farrugia, H. Yee and P. Nickolls, "Implantable cardioverter defibrillator electrocardiogram recognition with a multilayer perceptron," PACE, pp. 228-234, Vol. 16, No. 2, 1993

[5] F. Harrison, J. Marshall and L. Kennedy, "The diagnosis of heart attack: a neurocomputation approach," Computers in Cardiology, IEEE Computer Society, pp. 1-5, Vol. 1, 1991

[6] J. L. Holt and J. Hwang, "Finite precision error analysis of neural network hardware implementations," IEEE Transactions on Computers, pp. 281-290, Vol. 42, No. 3, 1993

[7] D. Rumelhart and J. McCelland, "Parallel distributed processing," MIT Press, Cambridge, MA, 1986

[8] K. Nakayama, S. Inomata, and Y. Takeuchi, "A digital multilayer neural network with limited binary expression," in Proc. Int. Joint. Conf. Neural Networks, pp. 587-592, Vol. 2, 1990

[9] S. Sakaue, T. Kohda, H. Yamamoto, S. Maruno, and Y. Shimeki, "Reduction of required precision bits for backpropagation applied to pattern recognition," IEEE Transactions on Neural Networks, pp. 270-275, Vol. 4, No. 2, 1993

[10] J. A. Swets, "Measuring the accuracy of diagnostic system," Science, pp. 1285-1293, Vol. 240, 1988

[11] A. F. Murray and P. J. Edwards, "Synaptic weight noise during multilayer perceptron training: fault tolerance and training improvements," IEEE Transactions on Neural Networks, pp. 722-725, Vol. 4, No. 4, 1993

[12] C. E. Cox and W. E. Blanz, "Ganglion-a fast field-programmable gate array implementation of a connectionist classifier," IEEE Journal of Solid-State Circuits, pp. 288-299, Vol. 27, No. 3, 1992

[13] N. M. Botros and M. Abdulaziz, " Hardware implementation of an artificial neural-network using field-programmable gate arrays(FPGAs)," IEEE Transactions on Industrial Electronics, pp. 665-667, Vol. 41, No. 6, 1994

[14] Xilinx, "The programmable logic data book," San Jose, USA, 1994

To appear in Proc. 1996 IEEE Engineering in Medicine and Biology 18th Annual International Conference, IEEE Press
Paper Number: 626

CLASSIFICATION OF BALLISTOCARDIOGRAPHY USING WAVELET TRANSFORM AND NEURAL NETWORKS

Xinsheng Yu, Don Dent and Colin Osborn
Faculty of Design & Technology, University of Luton, Luton LU1 3LF, U.K.

Abstract : Ballistocardiography(BCG) has an interesting feature that no electrodes are needed to be attached to the body during measurements. This provides a potential application to asses the patients heart condition in the home. Artificial neural networks(ANNs) have several properties that make them promising for the automatic signal classification problems. In the time domain of the BCG classification, the whole cardiac cycle of BCG waveform needs a large size neural network which makes the classification a computationally intensive task. By classifying the data in a compressed formal, savings in computer time may be realised. In this paper, we used wavelet multiresolution propriety analysis that allows to bring significant information content of BCG signal. Small subsets of the wavelet coefficients were used to classify the normal, hypertension and heart attack risk by neural networks. It is shown that the proposed system achieved overall 95.39% correct classification rate for testing data set. The classification of the features obtained by principal component analysis can only achieve 92.8% overall performance. The advantage of the proposed classification system is to be easily implemented in a portable device.

INTRODUCTION

Heart disease is one of the main factor causing death in the developed countries. Over several decades, variety of electronics and computer technologies have been developed to assist clinical practices for cardiac performance monitoring and heart disease diagnosis. However, the demands of practical health care requires that these technology are not limited to hospital environment but to be able to detect the risk of the heart disease in daily life in unrestricted condition. These requirements lead a increasing innovation in portable systems in reduction in size, weight and power consumption to an extent that the instruments are available in routine life monitoring and diagnosis to many users. BCG provides a potential application to monitor heart condition in the home because there is no electrodes required to be attached to the body during the measurements. Starr and Wood [1] shown that the BCG could be used to predict the development of ischemic myocardial disease. The prognostic value of the BCG in cardiovascular screening and epidemiological has also been reported [2], [3]. Recently, Zhao [4] studied chair BCG signals and indicated that the BCG has the value to predict the patients with mild hypertension who could be died due to myocardial infarction. However, all those reported works in the literature rely on the off-line analysis approaches of either visual interpretation of the BCG waveform or analysis the previous records on the digital computers with software packages, which are time consuming. It is hoped to develop a portable device for BCG signal processing so that the BCG can be used as an inexpensive and convenient screening test to help to diagnose the heart attack disease in their earliest and most treatable stages.

In this paper, we proposed an automatic BCG classification system based on the wavelet multiresolution analysis property and the learning capability of the neural network. The idea is to decompose the original signal into a series of views at different scales, then the coarse coefficients that have the significant information are presented to the neural network classifier. The proposed combined classification system can be easily implemented with today's available VLSI or DSP techniques for a portable device in real world applications.

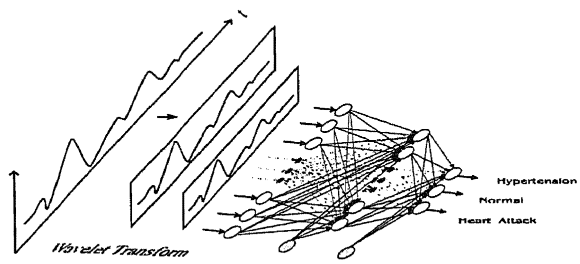


Figure 1: Combined Classification System Diagram

COMBINED WAVELET TRANSFORM AND NEURAL NETWORK CLASSIFICATION SYSTEM

The BCG data set used in this study was provided by the Medical Informatics Unit of the Cambridge University. The signals was sampled at 100 samples/Seconds from the seat of the chair. During the BCG recording, the reference ECG signal were recorded simultaneously form the arm of the chair at the same sampling rate. The R-wave of the ECG signal is used to identify each BCG cycle. The BCG signals are filtered and then averaged to reduce the background noise. Each averaged BCG signal was normalised into 80 points. Altogether, there are 58 normal subjects, 275 mild hypertension subjects and 6 subjects who died suddenly due to myocardial infarction within 24 months of the BCG recording. A classification of the whole BCG waveform can be computationally intensive by the ANNs. The high dimension inputs forces us to use a network with a higher number of free parameters which need a large training samples. Moreover, the presence of irrelevant feature combinations can actually obscure the impact of the more discriminating inputs. If the number of inputs to the ANN can be reduced without significant loss of information, then it should be possible to successfully classify the signal based on a reduced or compressed data set. A smaller data set would mean a smaller ANN and savings in time and hardware implementation complexity for the classification.

Mallat [5] show that the difference of information between the approximation of a signal at the resolution 2^{j+1} and 2^{j} can be extracted by decomposing this signal on a wavelet orthonormal basis of $L^2(R^n)$. In $L^2(R^n)$, a wavelet orthonormal basis is a family of functions which is built by dilating and translating an unique function. The idea is inspired by the fact that a prototype wavelet with its scaled and shifted versions used to express the original signal is usually an BCG like waveform. Thus, we decompose the original signal into a series of sub-signal at different scales, then the small approximated signal which still have the significant information were present to the neural network classifier. The proposed classification system is show in Figure 1. A fast wavelet transform program is implemented to decompose the BCG signal. The 12 coefficients of the impulse response derived by Mallat [5] is used in this study. A single hidden layer feedforward neural network trained with standard backpropagation learning rule was used. The hidden nodes were empirically defined as 4 hidden nodes. The 20 approximated coefficients were used as inputs to the neural network. The 3 outputs represent the normal, hypertension and heart attack subjects. The learning rate and momentum are empirically defined as 0.25 and 0.5 respectively. The whole BCG data set was partitioned into training data set and testing set. To get a realistic figure of the performance, a test data set which is not previously used in the training phase is used to estimate the performance. The classification results are shown in Table 1, where N represents the normal subjects, MP represents the mild hypertension subjects, and HAR represents the patients have the heart attack risk that causes death.

Table 1: Classification Results of wavelet coefficients

	Training Data			Testing Data		
Class	N	MH	HAR	N	MH	HAR
N	30	0	0	25	2	1
MH	0	111	0	2	159	3
HAR	0	1	2	0	1	2
		99.3%			95.39%	

To evaluate the performance of the proposed classification system, the conventional feature extraction method of principal component analysis (PCA) was used for BCG signal feature extraction. A neural network learning algorithm for PCA based on the modified Oja's rule [6] was implemented to extract the first 10 principal components. The combined PCA classification system consists two phases. First, the PCA network was trained with unsupervised learning BCG training set which the mean of the data was subtracted from each individual data vector, and then the extracted features were labelled and used to train a feedforward neural network with 10 inputs, 4 hidden nodes and 3 outputs. Once the training was finished, the unused testing data set was used to evaluate the performance. Table 2 shows the classification results.

Table 2: Classification Results of PCA features

	Training Data			Testing Data		
Class	N	MH	HAR	N	MH	HAR
N	29	1	0	25	3	0
MH	0	111	0	6	156	2
HAR	0	3	0	0	3	0
		97.2%			92.8%	

DISCUSSION AND CONCLUSIONS

The objective of this study is to develop a simple and reliable classification system which can be implemented into a portable device for real world application. The property of the wavelet decomposition of signals is to identify features such as wave peaks localised in time and scale. It is considered that this method is similar to the feature extraction method that take the wave peak, wave type and slop etc. into account. Thus, the wavelet transform provides a new approach in BCG data dimentionality reduction. Furthermore, the wavelet transform has a compact on the computation complexity.

Comparing to the conventional PCA method for feature extraction, the proposed classification system has demonstrated a good performance on the testing data set, especially on the heart attack subjects. On the other hand, the wavelet transform can be simply realised by finite impulse response(FIR) filters. These property make them suitable for real time application.

REFERENCES

[1] I. Starr and E. C. Wood, "Twenty years studies with the Ballistocardiograph, the Relation between the amplitude of the first record of 'health' adults and eventual mortality and morbidity form heart disease," Circulation, Vol. 23, 714-732, 1961

[2] C.E. Kiessling, "Preliminary appraisal of the prognostic value of ballistocardiography," Bibliotheca Cardiologica, Vol.26, 292-295, 1970

[3] T.N. Lynn and S. Wolf, "The prognostic significance of the ballistocardiogram in ischemic heart disease," Am. Heart J., Vol. 88, 277-289, 1974

[4] G. Zhao, "Development of New Ballistocardiographic System and Applications of Neural Networks to the Analysis of Ballistocardiograms," PhD thesis, 1994

[5] S. Mallat, "Multifrequency channel decomposition of images and Wavelet models," IEEE Trans. ASSP, Vol. 37, 2091-2110, 1989

[6] E. Oja, N.H. Ogawa, and J. Wangviewattana, "Principal component analysis by homogeneous neural networks, Part I, Part II," IEICE Trans. INF. & EYST., Vol. E75-D, 366-381, 1992

A Wavelet Multiresolution and Neural Network System for BCG Signal Analysis

Xinsheng Yu*, De-Jun Gong**, Colin Osborn* and Don Dent*
*Department of Electronics and Computer Engineering, University of Luton, Luton LU1 3LF, U.K.
**Institute of Oceanology, Chinese Academy of Sciences, 7 Nanhai Road, Qingdao, P. R. China

ABSTRACT: *Ballistocardiography (BCG) has an interesting feature that no electrodes are needed to be attached to the body during measurements. This provides a potential application to asses the patients heart condition in the home. Artificial neural networks (ANNs) have several properties that make them promising for the automatic signal classification problems. In the time domain of the BCG classification, the whole cardiac cycle of BCG waveform needs a big size neural network and a large training samples which make the classification a computationally intensive task. By classifying the data in a compressed formal, savings in computer time may be realised. In this paper, we used wavelet multiresolution analysis propriety that allows to bring significant information content of the BCG signal. Small subsets of the wavelet coefficients were used to classify the normal, hypertension and heart attack risk subjects by a single hidden layer neural network. It is shown that the proposed system achieved overall 94.66% correct classification rate for testing data set. The advantage of the proposed classification system is to reduce the computation complexity and to be easily implemented into a standalone device for real time application.*

1. INTRODUCTION

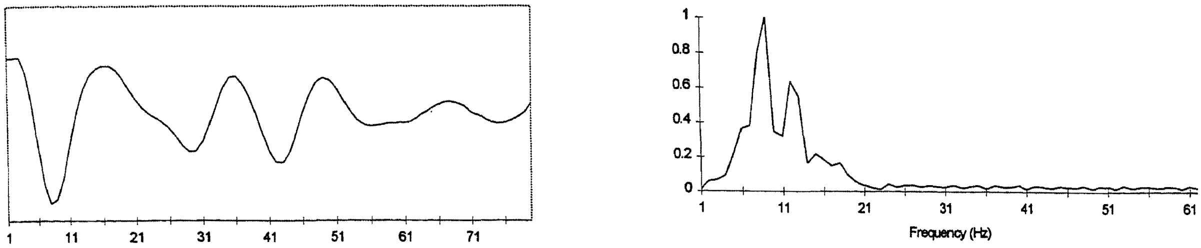
Heart disease is one of the main factor causing death in the developed countries. Over several decades, variety of electronics and computer technologies have been developed to assist clinical practices for cardiac performance monitoring and heart disease diagnosis. However, the demands of practical health care requires that these technology are not limited to hospital environment but to be able to detect the risk of the heart disease in daily life in an unrestricted condition, so the patients can receive the correct treatment at early stage. These requirements lead a increasing innovation in portable systems in reduction in size, weight and power consumption to an extent that the instruments are available in routine life monitoring and diagnosis to many users. BCG provides a potential application to monitor heart condition in the home because there is no electrodes required to be attached to the body during the measurements. Starr and Wood [1] shown that the BCG could be used to predict the development of ischemic myocardial disease. The prognostic value of the BCG in cardiovascular screening and epidemiological has also been reported [2], [3]. Recently, Zhao [4] studied chair BCG signals and indicated that the BCG has the value to predict the patients with mild hypertension who could be died due to myocardial infarction later. If these patients can be diagnosed in an early stage, the necessary treatment could be carried out to prevent the death. However, the reported works for BCG signal processing in the literature rely on the off-line analysis approaches of either visual interpretation of the BCG waveform or analysis the BCG records on the digital computers with software packages. The time requirement and cost for these methods are considerable. There is an obvious need to develop an automatic BCG classification system so that the BCG can be used as an inexpensive and convenient screening test to help to diagnose the heart attack disease in their earliest and most treatable stages.

In this paper, we proposed an automatic BCG classification system based on the property of wavelet multiresolution analysis and the learning capability of the neural networks. The idea is to decompose the original signal into a series of views at different scales, then the coarse coefficients have the significant information are presented to the neural classifier. The proposed combined classification system can be easy to be implemented with today available VLSI or DSP techniques for portable device in real world applications.

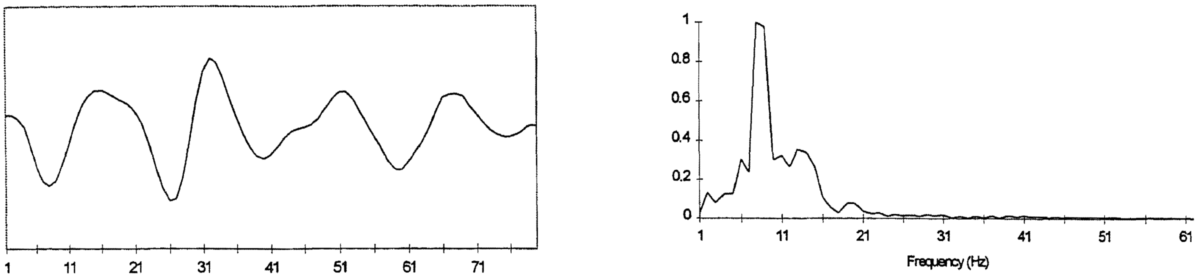
2. BCG SIGNAL PRE-PROCESSING AND DIMENSIONALITY REDUCTION TECHNIQUES

The BCG data set used in this study was provided by the Medical Informatics Unit of the Cambridge University. The signals was sampled at 100 samples/Second from the seat of the chair in a clinical trial. During the BCG recording, the reference ECG signal were recorded simultaneously form the arms of the chair at the same sampling rate. The R-wave of the ECG signal is used to identify each BCG cycle. The BCG signals are filtered and then averaged to reduce the background noise. Each averaged BCG signal was then normalised into a standard 80 points. Altogether, there are 58 normal subjects, 275 mild hypertension subjects and 6 subjects who died suddenly due to myocardial infarction within 24 months of the BCG recording.

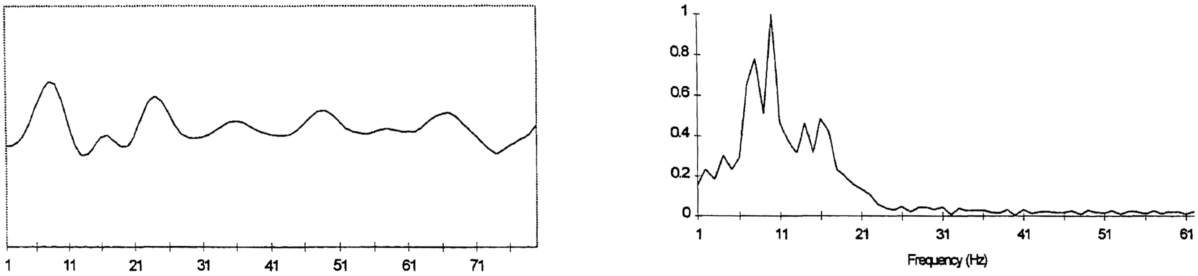
A classification of the whole BCG waveform can be computationally intensive by the ANNs. The high dimension inputs forces us to use a network with a higher number of free parameters which means a large



(A)



(B)



(C)

Figure 1: BCG mean of the normal (A), hypertension (B), and heart attack (C) subjects and its corresponding FFT.

training samples is required to estimate these parameters. The time involved in the network training is considerable. Moreover, the presence of irrelevant feature combinations can actually obscure the impact of the more discriminating inputs. If the number of inputs to the ANN can be reduced without significant loss of information, then it should be possible to successfully classify the signal based on a reduced or compressed data set. As can be seen form Figure 1, the spectra power are mostly within 20 Hz. A smaller data set would mean a smaller ANN and savings in time and hardware implementation complexity in real world application.

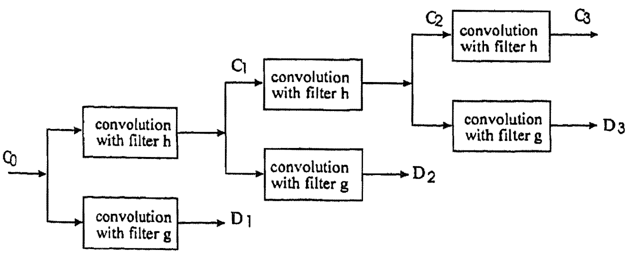


Figure 2: The multiresolution wavelet decomposition algorithm

Recently, the development of wavelet transform has introduced a new technique for various signal processing application [5], [6]. Mallat [7] shown that the difference of information between the approximation of a signal at the resolution 2^{j+1} and 2^j can be extracted by decomposing this signal on a wavelet orthonormal basis of $L^2(R^n)$. In $L^2(R^n)$. A wavelet orthonormal basis is a family of functions which is built by dilating and translating an unique function. The wavelet multiresolution is computed with a pyramidal algorithm based on convolution with quadrature mirror filters. Figure 2 shows the multiresolution wavelet decomposition algorithm. h represents low pass filter and g represents the high pass filter. At each pyramidal process, the signal $f(x)$ is decomposed into its coarse component $C_{2^j}f$ and detail component $D_{2^j}f$ at various scales 2^j. Signals at various resolution give information about different features. At a coarse resolution, we get information about the large features of the BCG wave; while at a fine resolution, we get information about more localised features of the single wave peaks. In the classification phase, not every resolution is equally important to provide the classification information. We can choose a proper resolution and discard the selected component without much signal distortion. Thus, we decompose the original signal into a series of sub-signal at different scales, then the small coarse components which still have the significant information are present to the neural network classifier. The filter used in this study is 12 coefficients derived by Mallat [7]. In this classification application, the signal reconstruction procedure is not required. Thus, only low pass filter processing is considered. A fast wavelet transform algorithm is adopted [9]. As the original data frame has 80 sample points, at each successive resolution level, the pyramidal algorithm compresses the data by a factor of 2. Figure 3 shows the three levels of the coarse coefficients. In our experience, the level two of 20 coarse components are accepted. This transform level has relatively good classification performance according to the preliminary results.

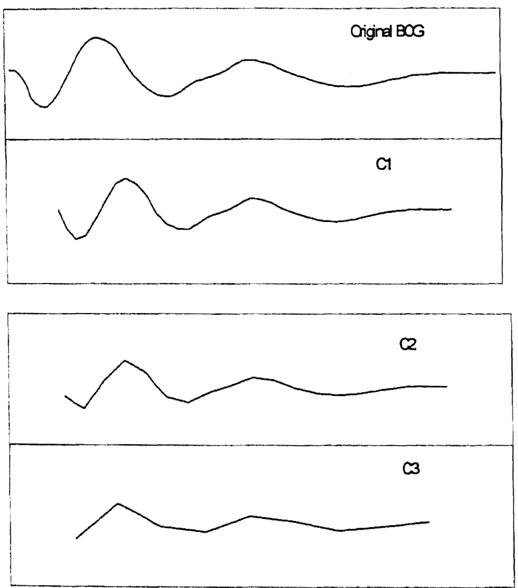


Figure 3: Coarse components of the BCG waveform

3. CLASSIFICATION RESULTS

To get a realistic figure of the performance, a test data set which is not previously used in the training phase is used to estimate the generalisation performance. Considering the limited patterns of heart attack subjects in the whole samples, we partitioned the whole BCG data set randomly into two data sets which shows in Table 1. One data set is used for the neural classifier training and another data set is used for test. We then swap the training data set and testing data set and repeat the training and testing procedure.

A single hidden layer feedforward neural network trained with standard backpropagation learning rule is presented for the separation the subjects. The hidden nodes were empirically defined as 4 hidden units. The 20 approximated coefficients were used as inputs to the neural network. The 3 outputs represent the normal, hypertension and heart attack classes. The learning rate and momentum are empirically defined as 0.25 and 0.5 respectively. The training process was stopped until no obvious MSE changes. Both training set and testing set are used to evaluate the network performance. The classification results are used to built the confusion matrix that the row represents the actual output of the network. The results of using data set 1 as training data set are shown in Table 2 and Table 3 illustrates the classification results of using data set 2 as the training data set.

Table 1: Partitioned Two Data Sets

	Data Set 1	Data Set 2
Normal	30	28
Hypertension	111	164
Heart Attack	3	3

Table 2: Classification Results of Using Data Set 1 as Training data

	Training Data			Testing Data		
Class	N	M	HA	N	H	HA
Normal (N)	30	0	0	24	3	1
Hypertension (H)	0	111		3	159	2
Heart Attack (HA)	0	1	2	0	1	2
		99.3%			94.87%	

Table 3: Classification Results of Using Data Set 2 as Training data

	Training Data			Testing Data		
Class	N	M	HA	N	H	HA
Normal (N)	28	0	0	27	3	0
Hypertension (H)	0	164		0	107	4
Heart Attack (HA)	0	1	2	0	1	2
		99.49%			94.44%	

4. DISCUSSION

The experiment results show that the overall 94.66% performance on the testing set was achieved. We then using data set 1 as training data to train a single feedforward neural network with 15 hidden units in the hidden layer. The results are illustrated in Table 4. Comparing the Table 4 with Table 2, the performance of combined wavelet transform and neural classifier system offer the better results than the single neural classifier especially for the heart attack subject. This might due to the fact that only small heart attack patterns in the training samples. In real world application, it is not realistic to obtain a large database. Therefore, it is desirable to reduce the input features to minimise the network size and to improve the classification performance. Wavelet transform provides a solution to this problem.

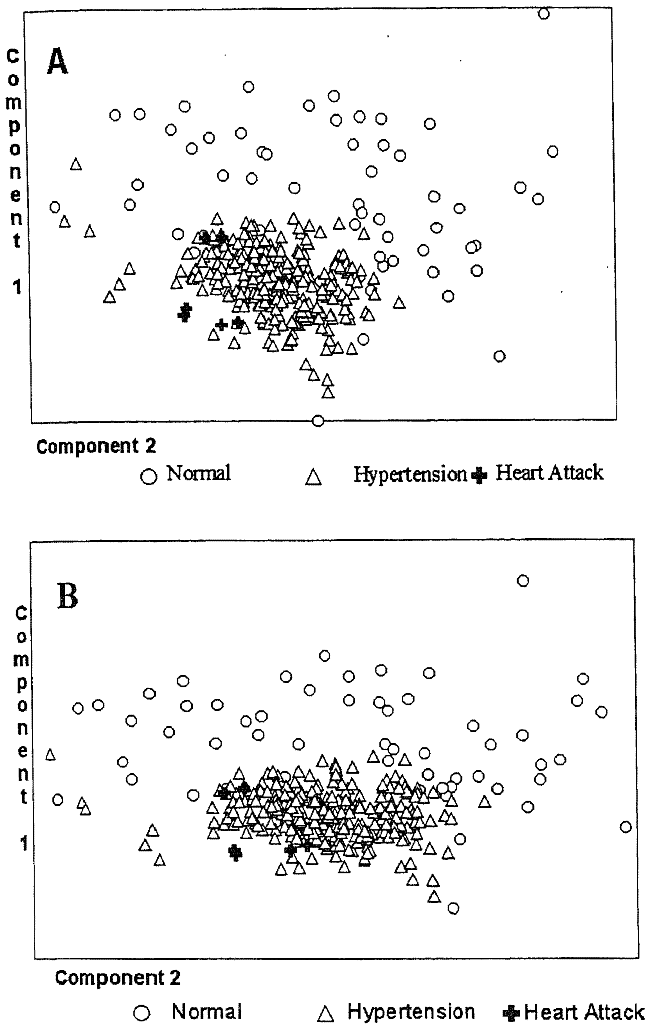


Figure 4: Two principal components map of (A) original BCG signal and (B) 20 coarse components.

The property of the wavelet decomposition of signals is to identify features such as wave peaks localised in time and scale. It is considered that this method is similar to the feature extraction method that take the wave peaks, wave type and slop etc. into account. Thus, the wavelet transform provides a new approach in BCG data dimentionality reduction. Furthermore, the wavelet transform has a compact on the computation complexity. To explore the decimated BCG signal properties, the first two principal components of both original BCG signal set and the 20 coarse components were extracted using the principal component analysis (PCA) network [8]. Figure 4 illustrates the 2D projection maps for these two data sets. By visualisation the two dimensional maps, we can observed that both map have the similar pattern distribution. The 20 coarse components are capable to reserve the most of the signal information as the whole BCG waveform has but with less dimentionality. On the other hand, the wavelet transform can be simply realised by finite impulse response (FIR) filters. These property make system suitable for real time implementation.

Table 4: Classification Results of Whole waveform Using Data Set 1 as Training data

	Training Data			Testing Data		
Class	N	M	HA	N	H	HA
Normal (N)	29	1	0	25	3	0
Hypertension (H)	0	111		5	155	4
Heart Attack (HA)	0	2	1	1	2	0
		97.9%			92.3%	

5. CONCLUSIONS

The objective of this study is to develop a simple and reliable classification system which can be implemented into a standalone device for real world application. It is well known that the neural networks have the good generalisation on the unseen data when the network are properly trained. Thus, the trained network architecture can be fixed into an automatic classification system for on-line application. A classification of whole BCG waveform needs a large network and large training samples to estimate the higher free parameters. In real world application, it is not realistic to obtain a large data samples. Thus, it is desirable to reduce the number of input features without significantly loss of classification information. We have proposed a BCG classification system based on the multiresolution wavelet transform and neural network. By using different resolution signal, the input dimension can be significantly reduced. The experiment results show that the proposed classification system have the better performance than a single neural classifier. As the wavelet transform can be simply realised by finite impulse response (FIR) filters, the computation complexity is significantly reduced. It is hoped that the true performance of the system will be assessed by a wide clinical trial.

6. ACKNOWLEDGEMENT

The authors would like to thank Dr R. Hanka form the University of Cambridge Clinical School for providing the BCG data sets and technical advises. We thank Dr. Zhao for valuable discussion. Thanks also to Dr R. Harvey, Dr. C. Crecraft and Dr R. Hearing for their encouragement and support. This research program was supported by the University of Luton.

7. REFERENCES

[1] I. Starr and E. C. Wood, "Twenty Years Studies with the Ballistocardiograph, the Relation Between the Amplitude of the First Record of 'Health' Adults and Eventual Mortality and Morbidity form Heart Disease", Circulation, Vol. 23, 1961, pp. 714-732

[2] C. E. Kiessling, "Preliminary Appraisal of the Prognostic Value of Ballistocardiography", Bibliotheca Cardiologica, Vol. 26, 1970, pp. 292-295

[3] T. N. Lynn and S. Wolf, "The Prognostic Significance of the Ballistocardiogram in Ischemic

Heart Disease", Am. Heart J., Vol. 88, 1974, pp. 277-289
[4] G. Zhao, "Development of New Ballistocardiographic System and Applications of Neural Networks to the Analysis of Ballistocardiograms", PhD Thesis, Cambridge University, Cambridge, 1994
[5] O. Rioul and M. Vetterli, "Wavelet and Signal Processing", IEEE Transaction on Signal Processing Magazine, Vol. 8, 1991, pp. 14-38
[6] I. Daubechies, "Orthonormal Basis of Compactly Supported Wavelets", Commu. Pure Appl. Math., Vol. 41, 1988, pp. 900-996
[7] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, 1989, pp. 674-693
[8] E. Oja, N. H. Ogawa, and J. Wangviewattana, "Principal Component Analysis by Homogeneous Neural Networks", Part I & Part II," IEICE Trans. INF. & EYST., Vol. E75-D, 1992, pp. 366-381
[9] M. A. Cody, "The Fast Wavelet Transform", Dr. Dobb's Journal, April, 1992, pp. 16-28